AD-A216 757

(2)

DTIC FILE COPY

US ARMY
LABORATORY COMMAND

TECHNOLOGY TO WIN

CONTRACT REPORT NUMBER 34-89
PREPARED FOR THE

HUMAN ENGINEERING LABORATORY

BY

Indiana University

W. W. Wright Education Building, 3rd and Jordan

Bloomington, Indiana 47405

ONLINE HELP SYSTEMS

Thomas M. Duffy
James Palmer
Susan Hathaway
Ann Aaron

September 7, 1989

US ARMY HUMAN ENGINEERING LABORATORY
HEL
ABERDEEN PROVING GROUND, MARYLAND

ABERDEEN PROVING GROUND, MARYLAND 21005-5001

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188
Exp Date Jun 30, 1986

1a REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b RESTRICTIVE MARKINGS
None

2a SECURITY CLASSIFICATION AUTHORITY

2b DECLASSIFICATION / DOWNGRADING SCHEDULE

3 DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution is unlimited.

4 PERFORMING ORGANIZATION REPORT NUMBER(S)

5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION
Indiana University

6b OFFICE SYMBOL (If applicable)

7a NAME OF MONITORING ORGANIZATION
U.S. Army Human Engineering Laboratory

6c. ADDRESS (City, State, and ZIP Code)
Indiana University
W. W. Wright Education Building, 3rd and Jordan
Bloomington, Indiana 47405

7b. ADDRESS (City, State, and ZIP Code)
ATTN: SLCHE-CS
Aberdeen Proving Ground, MD 21005-5001

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b OFFICE SYMBOL (If applicable)

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
Contract #DAAA1-86-K-0019AC85

8c. ADDRESS (City, State, and ZIP Code)

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
Online Help Systems (Unclassified)

12 PERSONAL AUTHOR(S)
Thomas M. Duffy, James Palmer, Susan Hathaway & Ann Aaron

13a TYPE OF REPORT
Final

13b TIME COVERED
FROM 6 Mar 86 TO 7 Sep 89

14 DATE OF REPORT (Year, Month, Day)
1989 Sept 7

15. PAGE COUNT
92

16 SUPPLEMENTARY NOTATION
This task was performed under the contract title: On-Line Help for Interactive Systems

17 COSATI CODES

FIELD	GROUP	SUB-GROUP
12	07	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
On-Line Help, Help Systems, human-computer interface, human factors; computer systems

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Computer systems were once used only by highly-trained specialists. Now, complex computer systems are used by people with various levels of skill and experience. On-line help must be designed so new or infrequent users can effectively use the systems without lengthy training or experimentation. The purpose of this project was to perform a systematic investigation of on-line help and to develop guidelines for the design of on-line help for computer systems. Three topics were investigated--

1. The help system should be arranged so that it can be efficiently maintained without sacrificing the capacity for many kinds of access. It should appear consistently so that entries can be added or amended easily. Alternate organizational strategies that balanced user-friendliness with ease of maintenance were developed and tested.

2. All entries in the help system should be structured for effective scanning and interpretation. Alternate presentation formats for on-line help information were evaluated. Optimal display strategies are reported.

(continued on next page)

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL

22b TELEPHONE (Include Area Code)

22c OFFICE SYMBOL

DD FORM 1473, 84 MAR
83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. (continued from page 1)

3. The type of information in the help system should be specific to the users for which the help system is designed. Procedures for assessing the kinds of information that will be useful were developed. Access channels to the on-line help database were designed.

ONLINE HELP SYSTEMS

THOMAS M. DUFFY, JAMES PALMER,

SUSAN HATHAWAY, & ANN AARON

SEPTEMBER 7, 1989

This work was funded by the U.S. Army Human Engineering Laboratory, Aberdeen Proving Ground, MD 21005-5001 under contract #DAAA1-86-K-0019-AC85.

Online Help Systems: An Introduction to the Issues

by

Thomas M. Duffy and James Palmer
Indiana University Apple Computer

7 September 1989

This work was funded in part by the United States Army, Human Engineering Laboratory, Aberdeen Proving Grounds, MD under contract # DAAA1-86-K-0019-AC85.

Online Help Systems:
An Introduction to the Issues

Thomas M. Duffy
Indiana University

James Palmer[1]
Apple Computer

The computer age is here! The information age is here! These are familiar, almost trite announcements. But what are the implications of the "computer age" for the designers of computer systems? And what in particular is the impact on the status of help systems? To answer these questions,we must start with a consideration of how the uses and the users of computer systems are changing.

Computers are becoming an integral part of virtually all aspects of our day to day activity. Services are being automated with computers. We see automatic tellers, computerized libraries, automated offices, computerized management of stocks, and computerized entertainment and eating recommendation services in hotels. In all of these, the individual operates a computer to obtain a particular service that otherwise would have been performed by a human.

Perhaps an even greater impact has been the use of the computer to complete tasks that we would have ordinarily done ourselves or have had to hire someone to do for us, i.e., personal computing. The first microcomputer, Kenbak I, was introduced in 1971. The first personal computer store did not open until 1975 and companies like Apple,Commodore and Tandy did not introduce their products until about 1977. Since that time we have seen an exponential growth in the sales of microcomputers. There were approximately a half million computers in use by 1978 and in just five years that number had grown 20 times to over 10 million computers. They project that this rapid growth will continue and they expect over 100 million computers in use by 1995.

The number of software packages, designed to do virtually anything we can imagine, also grew exponentially during this time. It is not only the breadth of applications but also the number of alternatives available for any application area. The March 1989 issue of PC Magazine reviewed 49 statistical packages, 9 scientific graphing packages, and 16 math toolboxes. That is a total of 74 different programs available for just three applications areas on just one platform, the IBM PC and compatibles. Add to that the Macintosh platform. MacWorld in February 1989 reviewed 8 spell checkers, 6 personal finance programs, 8 investment packages, and 5 tax preparation packages. Add to that the variety of drawing, word processors, music, payroll, database, spreadsheet, etc. programs and consider alternative platforms like the variety of workstations and mainframe, timeshare systems. The number of potential programs an individual may encounter is truly overwhelming.

This escalation in software packages is impacting not only end users but also programmers. There is an increasing emphasis today on cross platform connections, e.g., using Hypercard as a front end for a database application residing on a vax. Thus programmers must be be aware of many more platforms and of the strategies or capability for linking. Perhaps more importantly, however, is the proliferation of programming languages over the last fifteen years: C, ada, Lisp, Pascal, prolog, c++. small talk, t, scheme and a host of languages specific to particular applications, e.g., HyperTalk, database applications, and fourth generation languages. And, as with the applications, there are variations on all of these languages. For example, the February 1989 issue of Byte magazine reviewed 12 different C compilers.

We have gone on at length with these examples because we feel the consequence of this growth in both the breadth and depth of applications has tremendous implications for the development of user support systems.

Fewer Experts. The first implication is that there are fewer individuals available who are experts on any one program. In the past, it was likely that everyone at a given location used the same word processor, the same statistical package, and the same programming language simply because there were so few alternatives available. However, as the number of alternatives has expanded, and people are selecting applications based on their preferences, it is less likely that two people in the same room will be experienced with the same application. This means the most popular source of help is much less available. "Excuse me but do you know how to ..." is no longer an efficient help access strategy.

Fewer users developing expertise. Secondly, we generally will not develop and maintain expertise on most of the applications we use -- at least not the indepth, comprehensive knowledge we used to associate with "computer experts". In the early days of computers, the primary users were programmers. They programmed daily and generally used the same language and the same word processor to do it. Thus the continuous, frequent use led to expertise. However, many of the programs we use today are used only infrequently and for short periods of time. The sales person has to put together a presentation maybe monthly, or the secretary must work with the spread sheet only on Fridays, or the personal finance program is used only quarterly. Thus we never develop lasting expertise.

More transfer users. The third implication is that we are more likely to transfer to different applications in the same application area. It is not unusual to advance to a newer, better application than what you had been using . Nor is it unusual to change jobs or to have to work away from the normal facilities and find it necessary to use an unfamiliar product. Thus the number of transfer users -- people attempting to apply their knowledge of another program to this new program -- is increasing.

More novice users. Finally, the growth (both current and projected) in the number of computers in use means that there will be a continued growth in the number of first time computer users.

These changing characteristics of the user population have profound implications for the design of software packages. It is becoming increasingly important that software applications be easy to use. It is less likely that features and power alone will sell a product when there are typically several competing products with basically the same power. Usability is the key! Indeed, it has even been suggested that new, reasonably powerful, computers have failed on the market because they were not easily usable. It has also been suggested that some very simple database programs have very high popularity simply because they are easy to use.

Let us emphasize that we do not believe usability is an issue simply because of the large number of novice users. Usability is also an issue of increasing importance to people who consider themselves "expert" computer users. As we suggested in the first and second points above, these experts will still be using a wide variety of applications spanning many application areas. They are no different from anyone else in the frequency with which they may use a personal finance package or a tax preparation package.

Let us also reemphasize the number of alternative programs the expert has to use even in his or her area of expertise. We noted that there are over 49 statistical packages for the PC; that doesn't include the MAC or mainframe programs. Imagine a consultant on statistical analysis trying to work with this variety of programs if they were all as poorly designed and poorly helped (i.e., if they were as unusable) as the statistical packages of the early 1960s. Finally, the growth in availability and in the variety and power of peripherals and add on cards has led to greater power and versatility of the software. The number and sophistication of computing tasks that can be accomplished is escalating incredibly. Clearly the expert must deal with more diversity and complexity and will, of necessity, have less expertise on any one program even in his or her area of expertise.

Usable Systems

How do we make computer systems more usable in order to accommodate these changing user characteristics? The initial, and still dominant response of the industry, has been to focus on the software interface. The buzz words of the past were "user friendly" and "transparent". We were to make the capabilities of the program, as well as the methods for using these capabilities, fully clear to the user by carefully devising what that user sees on the screen. The idea was that the user should be able to intuit what can be done and how to do it simply by looking at the screen. As we shall discuss shortly, there has often been a naive expectation with regard to the level of transparency possible; a naivete that fails to recognize the importance of the users knowledge. Nonetheless, the focus on the interface has had a very positive impact on usability.

The shift in focus to interface design required a very significant reorientation in the industry. The goal in computing had always been to maximize power and capabilities. Software engineers focused on the elegance and power of the code they developed. While power and capabilities are still very important, this new set of goals -- interface goals -- means that the engineer may have to trade off capabilities or elegance for usability. They have to think about how the code will be presented to the user and how it will be used by the user.

The introduction of the Xerox Star (Bewley, Roberts, Schroit, and Verplank, undated) and Macintosh computers placed even greater emphasis on usability. The challenges of designing direct manipulation interfaces and the use of metaphors in the design of the interfaces even brought new software engineering prestige and excitement to the goal of usability.

Today the human computer interface is a well established discipline with a major annual conference (CHI or Computer Human Interaction) held under the auspices of the Association for American Computing Machinery. There are also innumerable guidelines available for the design of the interface. Perhaps the most extensive listing is the 674 guidelines presented by Smith and Moshier (1986) covering both hardware and software aspects of the interface. A sample of additional sources that provide a comprehensive consideration on the design of the software interface are:

- Gardiner and Christie (1987) a book presenting a cognitive perspective on the interface design and presenting 100 guidelines derived from that perspective.

- Apple Computer (1987) a book of guidelines for the Apple and Macintosh interface.

- Apple Computer (1989) a book of guidelines on the HyperCard interface design

- Engel and Granda (1975) a list of guidelines for the IBM interface.

- Jones (1989) guidelines are embedded in principles for designing the interface.

In addition to these "how to" books there are also several very important books and articles presenting conceptual or theoretical perspectives on interface design Norman and Draper, 1986; Shneiderman, 1986). These books and articles focus on general considerations for interface design -- interfaces for personal computing programs. There are also numerous books to aid in the design of specialized applications like instructional software (see, e.g., Heines, 1984; Hannafin and Peck, 1988).

Clearly, interface design is now receiving significant attention in the field. But, even with all of this effort, is a transparent interface an attainable goal? Steve Jobs, in describing the goals for the Macintosh, stated that he felt the interface design should be so simple that documentation was unnecessary. The need for documentation was viewed by Jobs, and is still viewed by many designers, as a failure to design a good interface. The interface should be the only help a user needs. In fact, the argument continues, any plan for documentation may negatively impact the interface design. Once the interface designers know that there will be supporting documentation they will have a tendency to sacrifice "transparency" for power or elegance. Thus the strong statement of the advocates of a transparent interface is that usability rests with the interface alone.

Is it possible to design an interface where all users can intuit all possible tasks and all of the operations necessary to execute those tasks? We think not! And we think not simply because how an interface is seen, how it is interpreted, how it is understood, depends on the knowledge the user brings to the situation. It is not just the factual knowledge about computing and about the topic area that the individual brings to the situation -- though variations in knowledge of these facts would be sufficient to make a transparent interface infeasible. It is also that experience in a knowledge domain and experience with a technology significantly impacts how you view the world: what you notice, how you interpret what you notice, and how you link things together to represent a situation or a goal. Thus even if the interface was simple and clear -- which is always a goal -- we may expect users with different knowledge bases to interpret the capabilities and perhaps even the procedures differently. (See Norman, 1988, for a more comprehensive discussion of how design must take into consideration the user's goal and the user's knowledge -- and some grand examples of failed designs).

Our point can best be illustrated by considering an example often used by those who propose that the interface should be fully transparent so that it may stand without documentation. They will point to the pencil as a technology with just such a transparent interface. The pencil does a great many wonderful things and we do not need support materials on how to use it. It is clear how to use a pencil just by looking at it -- or so the argument goes Thus the goal of these advocates is to design the interface for an application so that it is as transparent as the pencil interface -- thus obviating the need for any other usability support.

Let's take a closer look at the transparent interface of the pencil. First consider the basic pencil and basic applications of the pencil, analogous to the most basic computers and computer programs. Is the use of this instrument really without instruction and without assistance? Absolutely not!

We have actually received significant training and significant help in using the pencil. And the more sophisticated the pencil, the more help we have received. Children in school learn how to hold a pencil. Even in later grades there may be error correction when they hold it improperly. This includes not only the general holding of the pencil, but also holding it at the right angle and moving it in the right ways to form particular symbols. The training to form those symbols using the pencil technology is significantly different from how we would train the students to make those very same symbols using a typewriter or a keyboard. Thus the training is specific to the technology of the pencil. Indeed we would wager that you can also remember the laborious hours in school mastering how to use the pencil to make just the right symbols. Some of you may even have used templates to guide your initial practice.

We have all had to learn the functional difference between #2 and #3 lead pencils -- which pencil to use for which task. For many of us, we also had to learn that there was an important distinction between #2 hard and #2 soft. Frequently we had to learn that the hard way -- finding the #2 pencil we had was the wrong #2 and having to interrupt the task (usually a test) we were doing to search for the proper #2.

Of course, we have also had to learn the difference between pencils and pens (fountain and ball point). This included matching the proper instrument with the marking goal and with the particular surface to be marked (for example, consider the problems in using a hard pencil on a glossy surface, writing graffiti on the bathroom tiles using a fountain pen, etc.). The learning also included knowing the consequences and error correction procedures when the contents of one of these writing instruments got free.

Thus far we have only been dealing with the everyday use of pencils -- analogous to the most basic computer platforms and applications. These are well learned skills. However, let us now extend the technology, much as computer technology is being extended. Who would consider the use of calligraphy pens to be intuitive and not require any help beyond the natural interface? The same can be said about the selection and use of pens for graphic art -- knowledge of the technology of the tools is an integral part of the expertise of the artist.

There can be little doubt that the "transparent" interface of the pencil actually required a significant amount of training in pencil tasks -- including the selection of the appropriate tool for the job, the actual use of the tool to accomplish specific tasks, and troubleshooting when the tool is not functioning properly. However, the impact of using the pencil goes well beyond this development of fact knowledge. It is part of the technology of writing.

Writing allows us to form grapholects such as standard English (with nearly 2 million words) and to develop fully analytic and abstract modes of thinking (Olson, 1985; Ong, 1985). Being part of a literate society -- being experienced in the technology of writing -- has transformed our way of viewing the world.

So to, we may expect the computer technology to impact our representation of the world. For example, accountants who once balanced books using ledgers and adding machines can now interact with electronic spreadsheets. As a new tool, the spreadsheet requires the accountants to perform their original tasks in new ways; as the accountant gains experience his understanding of the accounting tasks will undergo transformations.

We have explored the technology of the pencil because it nicely illustrates the importance of knowledge in using the technology. The use of the technology required a knowledge base, required training, and it's continued use changed our way of thinking. If the use of the pencil and the technology of writing requires this training and has this impact on problem understanding and problem representation, how could we possibly assume that computer technology could be transparent. Certainly computer programs are growing in complexity and even now they amplify the complexity found in the technology of graphics and calligraphy -- for which extensive training and help is required. We find it difficult to conceive of an interface that is transparent to users with different conceptualizations of the tasks they are to perform. Surely there must be additional help available to link the user who has a different perspective of the task to the perspective exemplified in the interface.

In sum, we see little doubt that a user will require help in using a software package of any substance regardless of how much effort the designers went to to make the interface transparent. This is certainly not to deny the importance of the interface design. An easy to use interface is obviously a critical goal. It is simply that additional help is an equally critical goal.

Helping the User

What basis shall we use for classifying help systems? What are the basic types of help systems? Since the focus of this paper is on the design of help systems, our classification system should reflect basic differences in design requirements.

Help systems have typically been classified in terms of the skill of the user or in terms of the types of document developed (Schriver, 1989; Brockmann, 1986; Kearsley, 1988). We will discuss these approaches to classification later. For the moment., let us argue that the most useful classification of help systems -- from the rhetorical

Users' Goal	Medium of Delivery hardcopy	 online
I want to Buy it		
I want to Learn it		
I want to do it		*online help*

Figure 1: Classification of Help Systems

and design perspectives -- is in terms of two dimensions: the delivery medium and the goal of helping. Crossing these two dimensions results in a matrix containing six cells, each representing a different way of helping (See Figure 1). Each of these cells may be further analyzed into subsystems requiring different design strategies and this more detailed view will be our goal a little later when we focus on the cell labelled *online help*.

Dimension 1: Medium of Delivery.

Assistance to the user may be provided through a variety of media. Most commonly the contrast has been between online and hardcopy delivery, but there are also examples of video and audio tape delivery of information for software applications. With the growth of CD ROM capabilities we may expect the video and audio delivery to also be "online" in the near future. For now, however, we will not consider the audio and video issues, but will restrict ourselves to the more traditional and more readily available text and still graphic online information.

There are two reasons for considering delivery medium as a critical dimension for discriminating help systems. First, our research has indicated that there is a considerable difference between online and hardcopy in the approach to design. Content and design decisions that are affected by the delviery medium are a basic consideration throughout this paper The most fundamental differences in terms of the impact on design decisions include the following:

- the lack of permanence of the online display
- the restrictions in size of screen in the online display
- the inability to interact with the user and to provide dynamic, animated displays in hardcopy
- the restrictions on cross referencing or multiple presentations of information in hardcopy
- the different requirements for navigation aids in hardcopy and online, e.g., menu design and search mechanisms

The second reason for identifying delivery medium as a dimension for classifying help is because online help offers numerous advantages over delivery in the print medium and we expect the importance of those advantages to increase over time (Brockmann, 1986; Shneiderman, 1986; Walker, 1987; Duffy, Gomoll, Gomoll, Palmer, & Aaron, 1988). Thus, we anticipate a significant movement of help systems to online delivery. Just what are those advantages? Online delivery allows

Greater availability -- With networks and portable computers becoming more prevalent, we anticipate that it will become increasingly unlikely that adequate hardcopy documentation will be available for all software applications at all delivery sites. Online information can provide a reliable source of information for all software packages and on all delivery platforms.

Easier access -- Online, the system can provide mechanisms for efficient access to the relevant information, especially in cases where that information might span many volumes of hardcopy documentation.

More interaction -- Online, both the user and the system can interact with the information. For example, the system can use the state of the application (its current context) to determine what information to provide to the user, or a monitor capable of plan recognition could help debug a user's faulty or inefficient procedures.

High accuracy -- Hardcopy documents require much longer production cycles. As computer companies adopt shorter and more efficient software development cycles, the pressure to adequately document a product increases. The time it takes to produce a book after it is written – layout, formatting, and printing – becomes a bottleneck. Either manuals go into production well before products are stable (resulting in manuals that are inaccurate or incomplete) or a company incurs costs in order to make the necessary changes and to begin the production cycle again. This production bottleneck is minimized in online documentation, allowing for more accurate representation of the final software product. In addition to the initial production, documentation updates, i.e., version increments, are more easily accomplished for online documentation. Therefore, once again, more accurate information is more readily provided.

Low cost -- In general, online information is less expensive to store, reproduce, and distribute. This is partially a matter of the size and weight of a disk as compared to a manual. Of course, the reproduction process is also much more straightforward for electronic information than for print information.

Multimedia and AI -- Online information can exploit multiple media, such as video, sound, and animation, and can apply techniques from Artificial Intelligence (AI).

Of course, there are disadvantages to the online medium as well. First, with most computer systems, users cannot work in the primary application while using the online information. In contrast, users can work comfortably with an application while using a hardcopy tutorial or manual. Second, it is well documented that most computer displays diminish the readability of text and the legibility of characters, two factors which make reading from screens more difficult than reading from a book or manual (Muter, Latremouille, Treurniet, & Beam, 1982; Kruk & Muter, 1984; Haas & Hayes, 1987). In general, these types of problems are *technological*: and advances in computer hardware and software certainly will reduce or eliminate the difficulties.

There also remain *conceptual* problems. For example, the familiar strategies for navigating through books do not apply to online information (cf. Robertson & Akscyn, 1982; Elm & Woods, 1985). Users, therefore, must learn how to interact with the new medium. This is a problem that is not as easily overcome simply because the technology of hardcopy text is an integrtal part of our culture. However, well designed navigation systems can go a long way toward aiding the user in adopting the conceptual framework necessary for this new technology.

In discussing online vs. hardcopy delivery we must consider the issue of acceptance by the user. There is data and widespread belief that users reject online aiding. They simply do not want and will not use online information. Thus why should we bother considering online delivery? Why not just present everything hardcopy? We have two reactions to this proposition.

First, there are indeed situations where the user will clearly prefer hardcopy documentation. Of course hardcopy will be preferred and necessary when the computing power is not available, e.g., during set-up and when fatal crashes occur. Also, if the use of the information is going to involve reading for a long time it is far preferable to relax with a book in a chair rather than paging through a file. Finally, if there is studying to be done, again a significant amount of time being spent with a text, the user will once again prefer a manual. Finally, if the delivery system does not have windows available and the user must remember a lot of information in going from the aiding system to the application, then a manual will be preferred in 99% of the cases. (Though even here the efficient information search possible in online delivery could make it preferred over hardcopy for large systems, e.g., nuclear power plants.)

Our second reaction is that rather than simply abandoning online delivery because of user comments, the comments should be a basis for rethinking the design of online delivery to better meet the users need. The user may be rejecting online aiding because it is poorly designed -- not because of an inherent weakness. For example most of the preference data is from systems where windowing is not available. Who can blame the user for not being willing to use a system where getting information requires existing the application, entering the aiding system, getting that information, existing the aiding system, and reentering the application?

The technological problems are being overcome -- windowing is generally available and large screen monitors are more and more common. However, there is an additional problem: most help system are very poorly designed. We need to rethink the design process and the design principles for online aiding if we are to make online a preferred delivery medium.

The issues in design are perhaps best understood by reference to the history of hardcopy documentation. Originally the hardcopy documents were written by experts for experts. The goal was to document the system (Duffy, 1985). When the personal computing market suddenly grew there was no change in the design process or the design principles -- and users roundly rejected the documentation. Since then the development process and the design principles have evolved, improving the quality of manuals so both manufacturers and end users consider the manuals an asset to the software product.

Online help is still in its early stages and, like the early days of hardcopy documentation, still carries the vestiges of outmoded beliefs. While the understanding of the design requirements is growing, there are still many designers and developers who consider the development of online help to involve simply putting the manual online or providing some quick reference information. Fortunately there is a growing recognition that the design requirements for online presentation are different from those of hardcopy documents. The industry is now reexamining the design process to adjust to the new demands of creating online help, and they are searching for effective online help design principles. The move is afoot to improve the help systems and change the preference of the users.

Another reason to pursue online delivery is that, from our point of view, it is the only viable delivery system for the future. Many of the advantages of online delivery are outlined above. However, let us summarize what we see to be the four most important issues for the future.

- the technological advances are removing the barriers to online delivery and presenting enormous communication advantages for the online presentation, e.g., digitized video and intelligent systems. We will simply be able to communicate more effectively online.

- the proliferation of software and hardware products will make document management and storage of hardcopy documents virtually impossible for the end user. We might imagine by the year 2010 that a special room in the house would be required for documentation.

- the increased networking in which software is centrally stored and distributed around a facility and even to facilities across the country will make it difficult if not impossible to provide adequate documentation to every terminal or workstation.

- the increased cost efficiency of storing, distributing, and updating online materials (and the concomitant increase in the cost for hardcopy materials) will make online delivery the marketing choice.

In sum, medium of delivery is considered to be one of the two primary dimension for classifying aiding systems because the different media require fundamentally different design considerations and because the media differ dramatically in the delivery capabilities. We may expect user preferences for media to change as both the design and computing technology advance.

Dimension 2: Goals of the User.

Any classification of systems for aiding the user must focus on the goals of the user. After all, the goal of any aiding system is to meet the user's needs, i.e., help him achieve his goals. This dimension in the classification of help systems ensures that the the user's task has a sifnificant influence on the specification of the design and content of the help. Indeed, the specification of user goals as a dimension assumes that different user goals lead to fundamentally different aiding systems. We feel this is a reasonable assumption.

What kinds of information do users want or expect? What questions should information systems be able to answer? Note that we view the function of information with respect to the context of the use of that information, not the writer's intentions or the static description of its form (cf. Bethke, Dean, Kaiser, Ort, & Pessin, 1981). The core idea behind our effort is to match the information provided to users with the different kinds of knowledge that they require.

We can describe three user goals in terms of the expressed statement of need.

I want to *buy* it. The exemplar for information that meets this goal is the sales demonstration, sales book, and the specification sheet. The **audience** is usually prospective buyers. Their **goal** is to (perhaps) buy the product or to understand how they can use its services; ultimately, the information reflects a persuasive aim.

I want to *learn* it. The exemplars for this category include tutorials and guided tours (which may emphasize a successful first experience – "the affective response of the user" – as an adjunct to the goal of learning). The users' **goals** are to learn what they can do with an application and how they can perform some set of important or fundamental tasks. Constraints on attention generally limit this set of tasks to a group of basic skills, although more elaborate forms of instruction, much like a course or curriculum, also exist. From the user's point of view, however, the distinguishing feature is the goal of learning rather than performing. Typically, the **context** is a set of artificial situations constructed so that the user can practice using the application; the user is not actually performing real job tasks. The **audience** includes both the prospective buyer and the novice or transfer user.

I want to *use* it. The exemplars for this category include reference information and procedural information. Critically, the **context** of the request is the actual work situation. The user is trying to perform a task with the application. The users' **goals** are to overcome impasses that prevent them from proceeding on their task. The **audience** is all users, depending on the type of knowledge they require. For example, the novice typically needs task-oriented information, while the expert wants access to reference material.

Dimensions we did not include.

The reader may note that our classification of aiding systems does not include some dimensions that others have typically associated with such a classification process. In particular we did not include a dimension that describes the kinds of document nor did we include a dimension of user expertise. Let us consider each of these dimensions in turn.

Type of Document or Type of Information.

Redish (1987) and Schriver (1989) both classify types of aids by the types of documents. For example, Schriver describes four categories of "texts": tutorial, operations guide (procedures), reference, and quick reference. This list might be extended with other types of documents, e.g., the "open me first" document or the first experience document.

Implicit in a document dimension is a dimension of the type of information that is presented. We may think of four basic types of information: instructional, procedural, explanatory, and facts (or specifications). The match to documents is reasonably straightforward, at least in terms of the primary information element in a particular document.

We rejected this as a dimension primarily because it seems to focus on the wrong issue. It, in fact, seems to (if you will excuse the cliche') put the cart before the horse. That is, the analysis of user needs should lead to the specification of types of information and types of documents that will satisfy those needs. Thus, the needs of the user leads to a consideration of information elements and document types.

We should emphasize that there is not necessarily a one to one correspondence between user needs and information elements. Thus aiding "I want to do it!" may require facts, procedures, and maybe even some explanation. The selection and organization of information elements is determined by considering what is required to fulfill the need.

User Expertise.

Even more common than the classification by type of document is the classification by type of user. At the basic level this is a dimension that is identified by its extreme points -- expert and novice. Recently, however, an intermediate point, the transfer user, has been of focal interest.

Kearsley (1988) goes beyond levels of expertise and identifies three dimensions to which those levels may apply. In his "conceptual models of help" he indicates that users may be defined in terms of their expertise with the computer, with the particular task domain, and with the particular application software. If we just use the extremes (expert and novice) of each dimension, Kearsley's model would yield nine user types ("novice with the application, experienced with the computer and experienced with the task area" being just one type of user). Kearsley (1988) further suggests that "each of these types of users is likely to need slightly different types of help".

We certainly concur that knowing the skill level of the users and providing support consistent with that skill is critical. However, that is an information design issue along with many other information design issues. For example, we also have to be certain that the information is written at a level of syntactic complexity that the user will be able to understand and in a language he or she can understand. In sum, while it is a critical design issue --- and a major focus in any task analysis -- it is not a dimension along which to classify the goals of helping. It is a design issue that applies to every help system!

Beyond this basic conceptual disagreement with the definition of help systems in terms of user expertise there are several reasons for believing that it would lead to bad and impractical design.

First, it is not a system that takes into account the user's perspective. The very task of selecting the appropriate help system shifts the focus of the user's thinking from "how do I" or "I want to learn about" to "Let's see, how much do I know about this product, about computers, about this task domain?". Thus having different help systems based on expertise forces the user to shift task focus to one of selecting the appropriate help system.

Of course, at some point in the future expert systems might be able to reduce some of this burden. However, the expert system would still have to be able to identify the user's level of expertise (on three dimensions -- and maybe more) for the particular task she is working on. Additionally, the user, at some point, would have to identify her overall level of expertise on each of the dimensions -- so that the system has a beginning place for providing help. We question whether people can accurately assess their expertise even on one of those dimensions, much less on all three. (And we must presume accurate placement is necessary, otherwise there is no need for different help systems.)

Finally, help systems based on expertise seems to us to be impractical from a motivational point of view. While people may be willing to classify themselves as "beginners" we do not believe that they will select the beginners help -- especially if they are experts in another domain.

Second, the use of user expertise as a dimension is impractical from a development point of view. Kearsley's expansion from one to three dimensions of expertise illustrates the combinatorial explosion that is possible. Should experience with particular platforms be a defining factor, i.e., do we need a dimension of general computer expertise and then another dimension for experience with the particular platform? Certainly Macintosh experience is important if you are working on a Mac -- but so is the overall level of computer experience. Then again, we only considered the extremes of the dimensions -- wouldn't we need help systems for the intermediate level user? It would seem that this level is very important on each of the four dimensions. We have now grown to 64 different help systems!

The third problem we have with expertise as a dimension is that it classifies individuals rather than the individual's knowledge of particular parts of an application. Few people will be experts on all aspects of an application. The power and diversity of applications is such that in many cases the expert has not explored particular capabilities. Then again, new peripherals and boards often open new capabilities of existing applications. An experienced individual may well require a tutorial or procedural information on unexplored features of a program. Similarly the relative novice may develop proficiency for particular uses of the application -- she may become a power user. Her information goals in that domain may be similar to that of the more general expert. It is the user's goals rather than her overall skill level that is the determining factor.

Fourth, the use of expertise as a dimensions ignores the tremendous overlap in the goals of individuals with different levels of expertise and in the kind of information they search for. Duffy, Ackerman, Grantham, and Kelly, 1985) interviewed business users of microcomputers, system operators of minicomputers, and engineers using programmable controllers. The goal was to understand, in some depth, how people used system documentation. Duffy et.al (1985) classified the users into expert and novice users based on their relative amount of experience and training, though, in fact, none of the users were novice in the sense of having no experience.

As part of the interview, the users were asked to think back about their prior use of the manual and consider the kind of information they were searching for. In particular they were asked to classify their information searches into searches for facts (or specifications), procedures, or explanations. They were then asked to estimate the proportion of their searches that fell into each category. Interestingly, none of the users had any difficulty with the task: they had a clear sense of each of the three categories of information[1].

The findings, presented in Table 1, show that there are indeed differences in the information needs of experts and novices. However, most importantly from our perspective, there is also tremendous overlap. Experts do indeed look for explanation and they do look for procedures. Novices are not just searching for procedures, they too want facts.

---

[1] The documentation did not contain tutorials and thus Duffy et.al. (1986) could not ask the extent to which they looked for tutorials. .

Expertise

Information Platform	Expert fact	Expert proc	Expert explan	Novice fact	Novice proc	Novice explan
microcomputer	.36	.28	.34	.28	.40	.31
minicomputer	.33	.20	.47	.25	.42	.33
programmable controller	.40	.21	.39	.17	.57	.24

Table 1. Proportion of tasks devoted to searching for facts (fa), procedures (pro), and explanations (exp)as reported by more and less experienced users.

In closing this discussion, let us re-emphasize that understanding the user's knowledge is critical to the design of an effective help system. However, there are multiple user types asking the same questions for much the same purpose -- and there is a continuum of expertise. How one deals with those multiple users is a critical issue in the design of each and every help system.

Online Help Defined

Online help is the online delivery of performance oriented information. It is information presented online that is designed to answer the question, "How do I?" as illustrated in Figure 1. The difference between a learning-oriented aim and a performance-oriented aim is critical. We restrict our use of the term online help to systems that support performance. The user of online help is trying to complete some task in an application. The information required may be facts, procedures, or even explanation. However, because the individual is in the midst of the task, the information must be:

- targeted to the tasks
- written in a style that leads to efficient transfer to the task
- accessed efficiently

In contrast, online tutorials and training support the goal of learning. There is less urgency to the situation. The goal is to build generality and to have the information in long term memory. It is okay to have the learner use contrived learning tasks to illustrate a point or to facilitate learning.

In the following paragraphs we will attempt to address some parameters that frequently lead to confusion in distinguishing online help from other types of help.

Error correction vs error detection. Online help systems provide the user with information to allow them to continue on. Error notices presented in the application are not help.

User vs System initiated information. Online help may be initiated by the user or the system may detect an error and present information to correct that error. Either system would be classified as an online help system since the information is efficiently answering the question "How do I?". However, our focus is on the user initiated quest for help.

Primary vs secondary sources of online information. Online help is a secondary source of information. The information supports the use of a tool (the application) which itself is used to accomplish a primary task. In contrast online databases, e.g., an online encyclopedia, or databases of newspaper articles, research reports, etc., are not online help systems. These are sources of information directly applied to the solution of the real world task.

Online documentation vs online help. The distinction between online help and online documentation is one of design strategy. Both attempt to provide answers to the question "How do I?" and, with effective search tools and text (see, e.g., Walker, 1987), both can be efficient sources of information. The limitations of the delivery platform may well dictate the appropriateness of online help or online documentation for a particular application.

The Online Help Design Process

by

Susan Hathaway and Thomas M. Duffy
Indiana University

7 September 1989

This work was funded in part by the United States Army, Human Engineering Laboratory, Aberdeen Proving Grounds, MD under contract # DAAA1-86-K-0019-AC85.

The Online Help Design Process

Susan Hathaway and Thomas M. Duffy
Indiana University

Online help is now a common feature of most application software and the quality of the help system is even part of the advertisment of the strengths of the application. With the growing importance of online help, the development of the help information is typically no longer an additional (secondary) requirement of the documentation group and the design of the help interface design is no longer an add-on requirement for the application developers. Rather, we are increasingly finding a help design team assigned to the design of the database and the interface. The online help system is now a specific component of the overall development process.

While online help development is now a recognized part of the software development process, it is a relatively new function with little information available on the strategies for effective design and development. Because of the dearth of information on the online help design process, we undertook a study that would tell us how online help designers conceptualize the process, what tasks they say are involved in designing online help, and what salient problems they encounter at various stages of the design process. We did not directly observe the design process or try to get at the thinking processes that designers go through during the process; rather, we asked designers to tell us, in retrospect, what they did and what they thought went on during their last online help design project.

Our long term goal is to improve upon the design process. However, we must understand it before we can improve it. Just as an analysis of the user's tasks and performance aids us in better understanding what is needed in the online help system, an analysis of the tasks and performance of the designer of online help will help us to better aid and evaluate the design process. Without this kind of analysis, designers have little to reliably guide them through the design process.

It is apparent that online help designers are searching for guidance and feel a need to better understand the design process. The online help designers whom we asked to participate in a study of the design process expressed incredible enthusiasm for the study. Many of them stated that they wanted to participate in the study because they were looking for ways to characterize the process and to evaluate the strengths and weaknesses of their own design process.

Retrospective methods, such as the one we used, have some limitations (see Ericsson and Simon, 1984), but they allow us to most easily determine the mental pictures that designers have of the design process and thus allow us to examine the conceptualizations of a greater number of designers. Further studies in which the design process is observed as it happens will be needed to determine how closely the actual design process fits designers' models of the process and to find out what kinds of behaviors, information, and organization are needed for an efficient and effective online help design process. In this study, however, our goal was to produce an initial model that we could later test and that would assist online help designers in communicating among themselves and with others about the design process. An additional goal was to gather data on the demographics of the help design teams—the size, backgrounds of the members, organizational placement, etc.

The data was collected over a period of 6 months. There were five rounds of correspondence (questionnaires, ratings, and card sort tasks) all conducted through the postal service.

Who Participated

Participants in the study were 20 volunteers recruited by word of mouth or at a seminar on online help held at the Massachusetts Institute of Technology during the summer of 1988. They came from fifteen companies in the United States and Canada, including Ashton-Tate, Data General Corporation, Documentation Development Inc., IBM, Interactive Development Environments (IDE), Microsoft Corporation, Searchquest Information Services, Sun Microsystems, Texas Instruments, and WordPlay. (Several other companies were also represented; however, we did not receive permission to mention those companies). In spite of the informal way in which we recruited participants, we believe that, given the mix of large and small computer and computer software companies, we had a reasonably representative sample of online help designers. There was some attrition throughout the study; in the final round, we received responses from only 17 participants.

The participants had designed online help for a wide range of computer sizes (mainframe, micro, and mini) and software types (system, word processing, spread sheet, graphics, data entry, database, programming and applications development, hypertext, utilities). Each designer had been involved in the creation of between one and eight online help systems. Table 1 shows the number of participants who had designed a given number of online help systems. The mean number of help systems designed by a participant was three; however, eight participants had designed only one online help system.

Number of Help Systems Designed by Study Participants

Number Designed	Number of Participants
1	8
2	2
3	1
4	4
5	2
6	2
7	0
8	1

Mean = 3.05

Table 1. Number of online help systems designed by study participants.

Participants came from a variety of educational backgrounds. Eight of the participants had a bachelor's degree as their terminal degree, six had a master's degree as their terminal degree, and six had either a Ph.D. or an Ed.D. Thus, all of the designers had at least a college degree and over half of them had obtained a post-graduate degree.

Table 2 shows the subjects that participants studied in college and graduate school. As the table shows, the most common area of study was English or technical writing. This educational background was reflected somewhat in the job titles of the designers: Five participants listed "Technical Writer" as their job title, making this the most commonly held job title among the designers.

Educational Background of Online Help Designers

Area of Study	Number of Participants
English/Writing/Journalism	10
Social Sciences	6
Other Humanities/Liberal Arts	5
Education	3
Natural Sciences	2
Business	2
Library Science	2
Engineering/Computer Science	1

Table 2. Number of study participants with college or graduate degrees in given areas of study.

Other job titles indicated that the designers as a group have expertise in a wide range of skill areas and that they hold positions having a variety of levels of responsibility. As Table 3 shows, although the writing/documentation perspective was dominant, other participants were involved in programming, marketing, human factors research, training, and business management.

Job Titles of Online Help Designers

Job Title
Technical Writer (5 participants)
CBT Designer/Developer
Programmer Analyst
Systems Analyst (2 participants)
Instructional Designer
Manager of Electronic Documentation
Technical Staff
Staff Information Developer
Director of User Interface Research Group
Systems Technology Consultant
Project Leader
President of Company
Senior Research Librarian
Advisory Information Developer
Research Staff
Assistant Marketing Manager
Manager of Online Documentation

Table 3. Job titles of online help system designers who participated in the study. All titles held by one participant unless otherwise noted. Several participant held more than one job title.

The Online Help Development Team

The size of the online help development team ranged from one to ten, with a mean size of 4.3 (standard deviation = 2.7). The responsibilities and opportunities given the teams varied considerably across participants. In only four cases was online help developed concurrently with the application; in most cases, the online help development effort began somewhat after the application development effort was started or even after the application was finished. In a significant number of cases, there seems to have been little collaboration with application designers; eight out of 17 participants stated, for instance, that the help team had little or no input into the design of the help interface—that it had already been determined by the application developers. (See Table 4).

Portion of Help Interface Designed by Online Help Designers

Portion Designed	Number of Participants
All	2
Most	3
Some	4
Little	2
None	6

Table 4. Extent to which participants were able to design the interface of the online help system.

Application development and online help were almost always in different divisions of the organization and were administered separately. In nearly every case, communication with application developers was either through informal links across divisions or through a 2nd level manager. In only two cases did participants say that the application developers and online help developers worked together on a recognized project team. However four other participants indicated that there was a project manager linking application and online help development, suggesting that the different groups might have worked together on a formal or informal team.

In contrast, online help and hardcopy documentation were generally in the same division. And, in at least a third of the cases, online help and hardcopy documentation personnel either overlapped or were part of the same project team.

In general, then, online help and hardcopy documentation seem to be more closely linked organizationally than online help and application development. However, participants reported some interesting variations in organizational structure. In one case, hardcopy documentation and application development were in the same division and online help was in another. In another case, online help was split between two divisions, with the content of online help overseen by the documentation division and the online help software overseen by the application development division.

The organizational differences reported by participants may reflect the differing levels of priority given to designing a good online help system, differing views of what online help is and what it should be able to do, and/or differing views of how to most efficiently design software and documentation. In most cases, these organizational differences are not linked with differences in help team activities (e.g., how much of the help interface the online help team designed, how many prototypes they designed and tested, how many iterative tests they did of the product).

However, in all three cases where the online help team had only informal links to application developers in another division, it did not get to design the help interface (although one team was able to give some limited input to the interface designers), made an average of less than one prototype, and conducted only one test of the help system. In each of these cases, online help was considered "a quickie, get-it-out-the-door project," as one participant put it. The online help "team," consisting of one or two technical writers, was assigned the task of designing online help late in the application development process. The major activity of the online help team was, in these instances, to write the content of online help.

In contrast, other participants reported broader design activities and more product testing. Of the remaining participants, two-thirds said that their teams designed at least part of the interface. All of these teams constructed at least one prototype; seven constructed multiple prototypes. Finally, all but two of the teams conducted multiple tests of the help system, with four teams conducting eight or more tests.

Thus, while some teams are given the responsibility of designing the interface and the content, and go through many iterative cycles of synthesis, analysis, and evaluation, other teams are limited to a quick writing and implementation of help content for a finished or nearly finished application. These findings suggest that online help design for one company or project may be very different from online help design for another company or project in terms of the scope of the design process needed. Indeed, when we asked online help designers to tell us about the steps, tasks, or stages in the online help design process, we found that some designers think about the process in broader terms than others do.

Steps and Tasks in Designing Online Help

We wanted to determine how designers conceptualize the online help design process. What tasks or activities stand out for them as being important to the process? How do they mentally organize these tasks into groups of activities when they think about the design process?

In order to find out, we first asked participants to think about their last online help design project and list seven steps (plus or minus one) that captured for them the entire online help design process from beginning to end. Participants had half a page to elaborate on each step and describe what was involved in that step.

There was much variability in the steps or stages listed by participants. Some participants listed a broad range of activities while others focused on a smaller range of activities at various points within that broad range. In other words, some participants listed as tasks within a step what other participants listed as single steps or stages. We summarized all of the steps listed by all of the subjects and produced a set of steps which encompassed the entire range of activities mentioned by participants. These steps, which we will refer to as the "Retrospective Steps," are listed in Table 5.

Steps in the Online Help Design Process
(Retrospective Steps)

- Research and Review
- Project Management
- Create Design Specifications for the Content
- Create Design Specifications for the Interface and Functionality
- Prototyping
- Produce Help Text
- Implement Help
- Pre-release Testing/Quality Assurance
- Post-release Testing and Maintenance

Table 5: Steps in the online help design process determined from examining and summarizing participants' lists of steps and tasks.

We used the Retrospective Steps in later rounds to organize questions about the online help design process. The steps are listed in Table 5 in a sequence in which they *might* occur, but they do not necessarily occur in any given order and indeed a given "step" may be distributed across the entire process, e.g., as would occur if the design process was iterative. Although several of our subjects took a very sequential approach in listing steps (e.g., Step 4: Test‹Step 5: Revise‹ Step: 6: Test again), most subjects indicated that some steps might occur simultaneously and that it was common to repeat steps several times during the design process. This is consistent with the design literature in other fields.

In addition to examining steps or stages of the design process, we also looked for the tasks involved in the design of online help. From the elaborations and explanations that subjects wrote for each step they listed, we compiled a list of 66 tasks. We used these tasks in two ways in the following round.

First, we had participants do a card sort of the tasks. The card sort is a technique that helps us find out what kind of mental picture a person has of a process or of a body of knowledge. In this case, we wanted to get a picture of the design process by seeing how each designer broke the 66 tasks into groups that belonged together. In preparing materials for the task card sort for a participant, each task was printed on a separate slip of paper and the 66 slips of paper were shuffled into random order before being mailed to the participant. Participants sorted the tasks into piles according to what tasks they thought belonged together. They then labelled each pile.

We used cluster analysis to analyze the card sort data. Cluster analysis is a statistical technique which shows the relationships between items and gives an average grouping across subjects. The technique shows us how, on the average, the group of participants thinks about how the tasks go together to form steps and how the participants as a whole conceptualize the design process.

The number of groups of tasks per participant ranged from five to twelve, with a mean of eight. The cluster analysis resulted in nine groups of tasks, which we labelled descriptively. The Cluster Analysis Steps, along with a description of some of the tasks included in each step, are shown in Table 6.

Appendix A lists these steps along with the tasks that fell into each group. As with the Retrospective Steps, the Cluster Analysis Steps do not necessarily occur in a given sequence. Rather, these steps can occur in a variety of orders and may be repeated many times during the design process.

The Cluster Analysis Steps and the Retrospective Steps differ considerably, although both are based on the same broad range of tasks. We believe that the cluster analysis results give a more accurate view of the way online help developers conceptualize the online help design process. However, completing the cluster analysis proved to be a lengthy process and, in the interest of maintaining the momentum of the study for the participants, we did not want to delay sending out the next round until the cluster analysis results were ready and had been checked. We therefore used the Retrospective Steps to organize later rounds.

Steps in the Online Help Design Process
(Cluster Analysis)

Step	**Tasks Included in Step**
Administrative Planning	Project management and gathering of needed authoring and presentation tools.
Analysis for Design	Analysis of the design problem and its constraints; includes user analysis and task analysis.
Usability Test Plan	Specifying usability goals of the help system and identifying benchmark tasks and criteria for evaluating usability.
Content/Interface Design	Development of presentation strategies, design concepts, and design document.
Link to Application	Planning communication between help system and application, checking help topics against the application, and mapping help topics to the application.
Developmental/Iterative Testing	Prototype development and testing, user testing, and revision of guidelines and specifications based on results of testing.
Production	Writing and implementation.
Quality Assurance	Review and editing.
Testing and Monitoring of Product	Field evaluations of pre-release and released product, benchmark tests of final product, and planning for monitoring and updating final product.

Table 6. Steps in the online help design process, and a general description of tasks included in those steps, determined from cluster analysis results.

Importance of Tasks

In addition to having participants perform a card sort with the 66 tasks, we also had them rate the importance of each task. They rated each task on a nine-point scale, with nine being the most important. We asked them to think about importance in two ways: They could think of importance in terms of who they would assign to do that task or they could think of it in terms of the degree of quality control they would require for the task. Would they, for example, trust a novice to do this task? Would they require little or no review once the task was completed? Would they perhaps not even include the task in the design process? Or, at the other end of the scale, would they want an expert to complete the task? Would they require multiple reviews and testing?

Based on participants' ratings, we calculated mean ratings for each task. The mean ratings ranged from 5.28 (indicating a moderate degree of importance) up to 7.94 (indicating a high degree of importance). This indicates that all of the tasks are generally relevant to the design of online help. However, for some tasks there was wide variability in the ratings because some participants rated the tasks very important while others rated them unimportant. This probably reflects, again, the differing purposes and priorities assigned to online help in different companies.

The mean rating of each task served as the basis for ranking the tasks from one to 66, with one being the most important. In Appendix A, the numbers in parentheses following each task shows the rank of the task followed by its mean rating.

The ranking of the tasks suggests that online help designers tend to view the design of the content and interface as the most important step in the online help design process. This is not surprising since this is at the core of what the designer is trying to produce. The importance of this group of tasks is shown by the fact that the first, second, and fourth highest-ranked tasks are included in this group and by the relatively large number of tasks (six) in the group that are ranked among the twenty most important tasks.

More About the Online Help Design Process

Based on the card sort data and the rating of tasks, we get some ideas of how online help designers view the design process. Although this conceptualization is a composite of many designers' mental models of the process and is not likely to match anyone's model exactly, it is the best approximation of the designer's conceptualization of the design process.

As we stated above, we were unable to use Cluster Analysis Steps derived from the card sort data because of time constraints. Instead, we used the Retrospective Steps to organize the questions in later rounds.

In the final round, we asked participants to think about the last online help system they had designed and to (1) estimate the percentage of total development time they had spent on each step and (2) rank the steps from one to nine, with one being the most difficult step and nine being the least difficult. As Table 7 shows, producing the help text was, on the average, the most time-consuming step. However, there was also especially high variability among the responses for this step: Participants' time estimates for this step ranged from 8% to 65%.

Although producing the help text was considered the most time-consuming step, it was not considered the most difficult step. Creating the design specifications for the interface and for functionality was ranked as the most difficult step, followed by creating the design specifications for the content. As reported above, the card sort and task rating activities indicated that designing the interface and content was also the most *important* part of the online help design process. Again, not all participants agreed that these were the most difficult steps: Rankings for both of these steps ranged from one to seven.

Development Time and Difficulty of Steps in Online Help Design

Step	Mean % of Time	SD	Median Difficulty	Range
Research and Review	7.66	7.46	7	1-9
Project Management	8.06	5.09	6	1-9
Design Specs/Content	7.59	5.44	3	1-7
Design Specs/Interface	8.28	5.53	2	1-7
Prototyping	8.16	5.92	4	1-6
Produce Help Text	35.69	16.46	3.5	1-6
Implement Help	11.25	6.13	4	1-7
Pre-release Testing/QA	9.94	7.31	5	2-8
Post-release Testing	3.69	3.24	9	5-9

Table 7. Mean percentage of total development time spent on each Retrospective Step and median difficulty rating of those steps (1 = Most difficult).

In the final round, we also asked participants more about their activities and thought processes during each step of the online help design process. This information is organized below according to the Retrospective Steps.

Research and Review:

An important activity in this step is gathering information about potential users. We asked participants from which methods they obtained the most information about the potential users. As Table 8 shows, participants found indirect means of gathering information to be the most important. In this table and many of the following tables, there are more responses than participants. This is because, for many of the questions we asked, participants were able to check off or list more than one item.

The preference for indirect information shown in Table 8 may be due to several factors. First, the online help designers may be given little direct access to potential users. Second, designers may not feel it necessary to question or observe potential users. The broad, general information that the designers are likely to get from marketing and sales, and the detailed conceptual and procedural analysis that they get from task analysis, may be far more easily obtained and useful during the early stages of the design process than the information that designers would get from interviewing or observing potential users. Finally, designers may lack the resources (e.g., time and money) to obtain the information they need directly from potential users. They may decide that it is wiser to use whatever resources they could steer toward user testing in other ways or at other stages of the design process.

Methods of Audience Analysis
Used by Designers

Method	Number of Participants
Talk to those who have customer contact (e.g., marketing, training, sales)	8
Task analysis	6
Marketing/sales data	6
Feedback from user groups	5
Survey potential customers	2
Interviews	2
Focus groups with potential customers	2
Listen in on customer support phones for a similar product	1
Interview potential customers	1

Table 8. Methods by which online help study participants obtained the most information about potential users.

In general, designers were somewhat to very confident that they knew their potential users in terms of the tasks they needed to perform, their information needs, and their skill levels. On a scale of one to nine, with one being not at all confident and nine being very confident, designers' confidence ratings ranged from four to eight, with a mean of 5.5.

Project Management:

When asked what skills are the most important skills for at least one person on the online help team to have, participants responded as in Table 9. In general, analysis and production skills closely related to developing online help were mentioned most often, with more general skills and traits mentioned by fewer participants.

Important Skills on Online Help Design Teams

Skill	No. of Responses
Audience/User analysis (includes task analysis)	7
Design Skills (HCI, graphics, etc.)	6
Writing/Editing skills	5
Communication/Interpersonal skills	3
Technical knowledge/Programming	3
Creativity	2
Flexibility	2
Learn and work rapidly	1

Table 9. Skills which participants suggested as the most important skills for at least one member of the online help design team to have.

Create Design Specifications for the Content:

Almost all of the online help designers in our study were able to design most or all of the online help system content. Table 10 shows the methods and sources of information which participants said that they used for identifying and organizing the content to go into online help. User/task analysis was by far the most frequently mentioned method for identifying and organizing content. It should be emphasized again that little actual user contact or testing generally went into the user analysis; rather participants tended to use task analysis, general data about intended users, and their own intuition to guide them.

When asked to identify the sources of models or explicit principles that they used for designing and writing the content of online help, participants most often listed (1) the experience and intuition of themselves and others and (2) HCI principles and research. A complete list of sources is shown in Table 11.

Methods Used to Identify and Organize Online Help Content

Method	No. of Responses
User/task analysis	10
Objectives/Functionality specs	5
Written documentation	5
Engineering, technical staff input	3
Prior projects/versions	3
Interface, system constraints	2
Marketing information	2
Background knowledge of team members	2
Competitor's help products	1
Product features	1

Table 10. Methods and sources of information which study participants say they used to identify and organize the content of the online help system.

Sources of Models and Principles for Designing and Writing Content

Source	No. of Responses
Experience and intuition of self and others	7
HCI principles/research	6
Copied other products of previous versions	4
Marketing reference literature	1
Inhouse guide	1
Prime Computer Style Guide	1
Harless Performance Guild Workshop	1
Duffy's Evaluation of Online Help	1
Chicago Manual of Style	1

Table 11. Sources of models or explicit principles which participants said they used for designing and writing the content of online help.

The style and format principles or strategies that participants considered most important in designing the content are listed in Table 12. These principles and strategies are grouped into general writing and content principles, format principles, and access/navigation principles.

Style and Format Strategies and Principles

General Writing and Content Principles
Complete
Concise
Clear/Written for user
Consistent
Accurate
Procedural
Provide examples
Use present tense
Use 2nd person
Use Active voice

Format Principles
Organize and write modularly
Make each procedure and task self-contained
Chunk information usefully
Use consistent format
Keep interface as simple as possible
Proved alternatives to prose (i.e., graphics)
Leave plenty of white space

Access/Navigation Principles
No "go-to's"
Use different screen styles to aid navigation
Include headers to aid navigation
Tell how to get back to application
Give access to reference before procedural
Index by subject and by alphabet
Allow quick access

Table 12. Style and format principles or strategies that participants considered most important in designing the content of online help.

Finally, we asked participants what percentage of the content in their last help system was devoted to each of the following categories of information: Procedural, explanatory, tutorial and reference. We also asked what percentage they thought *should* have been devoted to these categories. Results are shown in Table 13.

For all categories, the range of responses was rather large, with the range being smallest for explanatory information. This provides further evidence that online help designers have different views of what online help is and should be.

Categories of Information in Online Help

Type of Information	% Devoted	Range	% Should have devoted	Range
Procedural	32	0-95	32	5-75
Explanatory	18	0-50	20	5-40
Tutorial	11	0-60	11	0-40
Reference	39	0-90	35	10-80

Table 13. Mean percentage of content in last help system which was devoted to given categories of information and participants' judgements about what percentage *should* have been devoted to each category.

When each individual's estimate of what percentage of content *was* devoted to a given category of information was compared to his or her judgement about what percentage *should* have been devoted to that category, the number of participants who said that a *smaller* percentage of content should have been devoted to the category (usually five people) was about the same as the number of participants who said that a *larger* percentage of information should have been devoted to that category. The exception was for the reference category: Only three participants felt that a larger percentage of content should have been devoted to reference information while seven participants felt that less should have been devoted to that category.

Create Design Specifications for the Interface and for Functionality:

Whereas almost all participants were able to design most or all of the content for online help, only two-thirds of the participants were able to design part or all of the interface; six participants did not design any part of the interface. It should come as no surprise, then, that when we asked participants to rank sources of interface design specifications in the order of how important the sources were in influencing the design of the participants' last online help systems, the most important source turned out to be pre-specifications made by the application designer. Other important sources are shown in Table 14.

Determinants of Interface Design Specifications

	Median Rank
Pre-specified by application designer	1.5
Analysis of audience and its use of the application	2
Adaptation of model in existing systems	2
Guidelines in house book	4

Table 14. Participants' rankings of the importance of determinants of design specifications for the interface in their last online help system
(1 = most important).

Prototyping:

We asked participants whether they constructed any prototypes of the online help system and, if so, how many and of what type. Two participants did no prototyping, fourteen participants constructed one or more online prototypes, and five participants constructed one or more paper prototypes. In all, eight participants constructed multiple prototypes (paper and/or online). Table 15 shows how much of the help system participants say they prototyped.

Amount of Help System Prototyped

Amount	Number of Participants
All	4
Most (75-99%)	2
Some (25-74%)	2
Little (1-24%)	7
None	2

Table 15. Amount of the help system prototyped by participants during the development of their last online help systems.

To evaluate the prototype, participants most often used team evaluations or review and evaluation by other departments in the company (e.g., research and development, hardcopy documentation writers). Eleven of the participants said that they used this method, while eight conducted user testing and two used software to check for problems.

Produce Help Text:

The tasks involved in producing the content were defined and distributed across team members in several ways. The most common way was to divide the tasks up by the type of information (reference, examples, procedures). Other methods mentioned included dividing content according to the writers' subject matter expertise, by team members' writing skills, by user tasks, by help screens, and by content topic areas.

The most common method of assuring the quality of the writing was editing. However, about a third of the participants stated that the writing was also given a technical review to ensure its accuracy. In one case, user testing was conducted. The team was responsible for quality assurance for the writing in about half of the cases; external editors or the company's quality assurance department was responsible in the other cases.

Implement Help:

We asked no detailed questions about this step.

Pre-release Testing/Quality Assurance:

Table 16 shows the number of iterative tests conducted of participants' complete help systems (i.e., after the systems were functioning) before the products were released. In more than half of the cases, only one or two tests were performed.

Number of Iterative Tests of Help System

Number of Tests	Number of Participants
1	5
2	4
3	3
≥ 8	4
Don't know	1

Table 16. Number of iterative tests of the complete help system conducted during the development of participants' last online help systems.

Testing was performed by a variety of groups in different companies. Table 17 shows who did the testing of participants' online help systems.

Testers	Number of Iterative Tests of Help System Number of Participants
Online help group	6
QA	6
HCI/Usability group	3
Application developers	1
Other	4

Table 17. Groups responsible for testing the help system during the development of participants' last online help systems.

Table 18 shows the kinds of methodology used for quality assurance/testing/review of the content and the interface of the help system. Less than half of the online help systems were evaluated by user testing, but in some other cases team members and other company personnel served as usability test subjects.

Methodology	Methodology Used in Testing of Help System Content (# of participants)	Interface (# of participants)
Review	13	6
User test	7	6
Team/QA use	4	5
Compare to manual	3	0
Lab tests	1	1
User questionnaire	1	2
Compare to corporate standards	0	1
Hunches	1	0
None	0	4

Table 18. Methodology used to test content and interface of complete help system during the development of particpants' last online help systems.

In general, participants were somewhat doubtful that they had tested adequately and were releasing an effective online help system. Three participants stated that they had many doubts, seven had some doubts, five had a few doubts, and two had no doubts.

Post-release Testing and Maintenance:

Nine of the participants said that their company has a mechanism in place for the post-release monitoring of products. Table 19 shows what kinds of mechanisms are in place in these companies.

Mechanisms for Monitoring of Released Help Systems

Mechanism	Number of Participants
Customer support group	4
Distributors, sales people	2
Letter from customers	2
Registration cards/Feedback forms in documentation	2
Hotline	1
QA department	1

Table 19. Mechanisms in place for the post-release monitoring of problems and updates of online help systems.

According to participants, the most important source of post-release feedback is information from marketing, training, product development, and customer support. Table 20 shows the sources of feedback ranked from most important to least important.

Sources of Feedback About Released Help Systems

Source	Median
Information from marketing, training, product development, customer support	1.5
User groups	2
User questionnaires	2
Information from product development	2
Applications information hotline	3
User tests with actual customers	3.5
Distributors, sales	3.5
Personal interviews with customers	4

Table 20. Rankings of importance of sources of feedback about release product (1 = most important). Participants ranked only those sources they received.

Summary of Design Process Findings:

The data from the online help designers indicates that each designer views the process somewhat differently, perhaps depending on the organizational context in which he or she works. Some designers tend to view the development process more broadly than others. Many indicated that the process involves iterative cycles of synthesis and evaluation and some mentioned analysis as part of this iterative cycle, as well. These results are compatible with the results of design studies in other fields.

Producing the help text was considered the most time-consuming step, overall. However, participants considered designing the interface and content to be the most important and difficult part of producing an online help system.

Designers often used experience, intuition, and indirectly gathered information about potential users to make design decisions. Less than half of the participants supplimented this with feedback from naive potential users at any given stage of the design process. We suspect that online help is given very low priority in some companies and that many of the participants lacked the resources to do user testing of the help system during and after development.

Lack of resources may also have been a factor in the limited extent to which participants were able to construct and test prototypes of the online help system. Less than half of the participants constructed multiple prototypes.

Whereas most participants were able to design much or all of the online help content, our results indicate that online help designers are less likely to have substantial involvement in the design of the online help interface. Indeed, one-third of the designers in our study did not design any part of the interface. The interface is often prespecified by the application designers, who are generally organizationally removed from the online help designers. Furthermore, application development is often well under way or even completed before online help design begins, giving online help designers even less opportunity to have an impact on the design of the online help interface.

The above discussion indicates that inadequate organizational support may be a problem for online help designers in some companies. This hypothesis was confirmed when we asked designers about the problems they have in designing online help.

Problems in Designing Online Help

In addition to finding out what is involved in designing online help, we wanted to know what kinds of salient problems online help designers encounter most frequently. To do this, we listed the Retrospective Steps and asked participants to accept those steps, for the time being, as steps they would go through in designing online help. We then asked them to think about their last online help design project and tell us the most serious problem that they had encountered in performing each step.

We also asked participants to list the strategy or strategies which they used to avoid or cope with each of these problems, but many of the strategies that were suggested tended to be trivial (e.g., "Plan for this.") so we did not use them in later rounds. It is likely that the detailed, focused problem-solving behavior that designers must use to avoid and cope with problems cannot be captured in a retrospective report such as this one.

From the participants' lists of problems, we complied all of the problems into a list of 71 "most important" problems. In the next round, we listed the problems under the Retrospective Steps in which participants said they occurred and had each participant rank order the problems in each step. The problems are listed under their steps in rank order in Appendix B, with 1 being the most highly ranked problem. The numbers in parentheses after each problem indicate that problem's median ranking and the range of rankings which participants gave the problem.

Four problems occurred across several steps: Adapting to change, dealing with incomplete information, understanding the user, and getting needed resources. The first two problems are unavoidable during the design process given the nature of design. Because there is always more to know about the design problem, information is always incomplete and as more information is gathered, changes in the design will be necessary.

However, these two problems are aggravated when the application for which help is being developed is undergoing changes at the same time that online help is being designed. Although concurrent development of the application and online help offers the designer many opportunities to design both a better application and a better online help system, dealing with the many adjustments caused by changes in the application can be frustrating as well as an information management nightmare. Furthermore, if inadequate time and other resources are allocated to designing online help, hurried evaluation at the end of the development cycle may indicate that major changes need to be made when there is not enough time to make those changes.

The third of the four problems—understanding the user—is unavoidable giving the nature of designing something *for someone else*. Because of what they learn in the process of designing online help, online help designers become less and less like the naive users who will interact with and react to their designs. Because of this increasing disimilarity to users, and because of the many and broad differences among users, designers must make a special effort to see that the online help system addresses the needs of naive users.

The above problem, though unavoidable, is manageable given the resources to survey and test users as needed. However, the fourth problem that online help designers commonly had was getting needed resources. Because management often viewed online help as unimportant and easily developed from hardcopy documentation, participants sometimes had difficulty getting the time and human resources needed to develop and test prototypes and conduct adequate pre-release testing. Thus, the lack of resources magnified the problems that designers had in understanding the user.

Although we did not generally look closely at the strategies suggested for avoiding and coping with problems, several strategies emerged again and again for coping with the problems of dealing with new information, with change, and with inadequate resources. First, participants suggest that it is worthwhile to keep asking for what is needed in terms of time and other resources and to point out the consequences to managers and other decision makers when these resources are not allocated. Second, they say that good communication skills (e.g., persuasiveness and making your needs known) are needed to deal effectively with the many disputes that come up during the design process and to get necessary assistance from team members and application developers. These strategies suggest that interpersonal skills may play a key role in avoiding and coping with problems in the online help design environment.

In order to cope with the problem of understanding the user when there are inadequate resources for user testing, online help designers probably tend to rely on intuition, research, and the information that they obtain from task analysis, marketing and sales data, and other front-end analysis activities to give them the user perspective needed when they evaluate online help. They may also conduct usability testing with relatively naive colleagues and friends who may or may not have much in common with intended users. One common strategy is to have people on the team who have not designed or used a given portion of the help system serve as evaluators for that portion. The idea behind this strategy is that the person who has not designed the online help interface, for example, is a relatively naive user of the interface, even if she has written the content for the online help system.

Improving the Design Process

We have looked now at how online help designers think about the design process, at what tasks they say are involved in the process, and at a few of the problems they typically encounter in designing online help. Our study does not provide observational data about how the online help designers actually design or what their thinking processes are during the design process; nor does the study give us any indication of what makes a design process more effective or less effective. This data will have to be collected in further studies.

However, the data from our study does show us some of the kinds of tasks designers must complete in designing an online help system and suggests that online help designers act similarly to designers in other fields. And, while we still have much to learn about what makes an effective design process, there are studies in the area of software design that offer suggestions appropriate to the online help design environment.

Olson (1985) describes how careful planning of prototype testing and analyses of testing results led to the design of a more usable interface. Such user testing not only added to designers' confidence that their product was good, but also gave them the data needed to convince others (e.g., developers and project managers) of the merit of their design. Olson suggests testing pairs of prototypes, the pairs differing in only one component of the design. Designers should choose this component based on whether it is important to decide about that component early in the design process, whether something useful can be learned from testing that component, and whether the component is likely to appear in future designs.

Good, Whiteside, Wixon, and Jones (198?) stress the importance of iterative design in creating a more usable interface. They state that:

> "Ordinarily, one writes some sort of specification before any code is produced. This specification represents a "commitment" by engineering of what they intend to build. If one could reliably extrapolate this specification into a working system, mentally envision people using it, and thereby successfully anticipate all the problems that users would face, this process would be simple, cost effective and efficient. Sadly, experience has shown that anticipating problems from specifications is not successful. An approach which has proven more successful is to develop a prototype early and assume that it will be changed." p. 146

It is especially important, they say, recognize the tentative nature of the initial design and to avoid commitment to a given interface. Commitment must be to functionality and levels of usability, not to a specific solution.

The above studies suggest that a focus on users and the tasks they must perform, and a commitment to iterative design and careful user testing, may vastly improve the usability of software. These suggestions are echoed by Gould and Lewis (1985), who report that although these principles are useful, they are most likely not obvious or used by many designers. They asked 447 systems planners, designers, programmers, and developers to write down the sequence of five or so steps one should go through in developing and evaluating a new computer system for end users. Only 62% mentioned early focus on user, only 40% mentioned user testing and empirical measurement of usability, and only 20% mentioned anything related to iterative design.

Appendix A: Cluster Analysis Steps and Tasks

Administrative Planning

Select or develop the required run time presentation tools (rank=7, mean=7.50)

Select or develop the required authoring tools (13, 7.33)

Specify the requirements for run time presentation tools (16, 7.28)

Identify operating system constraints on the interfaces (44, 6.56)

Specify the authoring tools that will be required (46, 6.50)

Determine the relative costs/problems associated with alternative technical solutions (49, 6.44)

Assign responsibilities: determine who will lead various efforts (55, 6.39)

Get resource requirements (time, money, personnel, etc.) filled (56, 6.28)

Timeline and resources: Identify the resources that are available and the constraints (time, money, personnel, design, etc.) on development; prepare a milestones chart (57, 6.28)

Work with software engineer to establish roles in implementing online help (64, 5.83)

Analysis for Design

Analyze users: Range of abilities (reading, analytical skills), needs, expectations, computer literacy. See what users of earlier versions or of similar applications need and want in the way of online help, including delivery preferences (rank=6, mean=7.56)

Prepare a strategy document defining audience, scope, and high-level design (9, 7.44)

Meet with developers to review the software resources available in the delivery systems and to determine how help can interface with the application, e.g., context sensitivity, how help can appear in relation to application, potential navigation strategies (12, 7.33)

Establish intent of help, e.g., aiding novice and other users, teaching, prompting, and/or serving as reference (20, 7.17)

Conduct user task analysis to determine all the tasks the user will perform (26, 7.06)

Meet with developers to review hardware resources available in the delivery system to determine how help can interface, e.g., type of CPU, color attributes, amount of memory and disk space, sound capabilities, etc.Define linkages between online help and various states of the application (31, 6.83)

Determine needs for practice/remediation/tutoring in the help system (37, 6.61)

Interview developers, marketers, and key decision makers to determine the functionality goals for the product, the characteristics of the intended audience, and the intent and purpose of the help system (47, 6.50)

Create a feasibility specification that gives a realistic appraisal of what the product and what the help system can realistically be expected to do (50, 6.44)

Establish the relation of online help to other information sources (e.g., written documentation) (52, 6.39)

Determine what platforms, peripherals, and other software will be used with the application (53, 6.39)

Identify and analyze any other documentation related to the application (e.g., marketing plans, user manual, earlier versions, related applications) (63, 5.89)

Usability Test Plan

Identify benchmark tasks and usability criteria for assessing help system usability (rank=18, mean=7.22)

Specify usability goals and measurable objectives that can be applied in assessing the usability of the help system (22, 7.11)

Content/Interface Design

Create guidelines for the design of the interface: access, exiting, navigation, menus, screen layout, etc. (rank=1, mean=7.94)

Establish design concept for the content: Outline the flow of help, identify linkages between topics, etc. (2, 7.89)

Prepare design document: Details on how to organize the help information, specification of what's in help vs. what's not, etc. (4, 7.72)

Establish design concept for interface: Storyboard part of the system to illustrate screen, layout, navigation, levels of help, etc. (8, 7.44)

Create guidelines for the content: Voice (e.g., first person), comprehensibility, level of detail, breadth of coverage, etc. (11, 7.33)

Select dialogue style(s) and presentation strategies for the help system (19, 7.17)

Develop templates/grids for online help screens/windows (30, 6.83)

Plan for use of graphics in online help screens/windows (38, 6.61)

Develop an instructional strategy for the help system (40, 6.61)

Determine style and other constraints that are set by company policy (65, 5.56)

Link to Application

Define interfaces (e.g., parameters passed) between the application and the help software (rank=14, mean=7.33)

Work with the application to develop a user perspective, identify functionality, identify possible metaphor for use in documentation, and identify problem areas (15, 7.33)

Define linkages between online help and various states of the application (28, 6.89)

Map help topics to the application (39, 6.61)

Outline the content of online help by topics/procedure (42, 6.56)

Document problem areas in the application where the user might get lost (43, 6.56)

Developmental /Iterative Testing

Create a prototype of part of the online help system and user test it (iteratively) (rank=3, mean=7.78)

Model/prototype two or more help systems and test and evaluate to determine which is best (27, 6.94)

Conduct user tests of the draft of the online help (32, 6.78)

Revise the guidelines/specs for the design of the content and/or the interface (45, 6.56)

Production

Write online help text being sensitive to the users' need to "get out of their mess" and being certain to provide step-by-step information (rank=17, mean=7.28)

Integrate prototype of help system with the application code (25, 7.06)

Link help into the application (33, 6.72)

Coordinate with any application development and/or hardcopy documentation efforts (35, 6.61)

Review the application periodically to see if and how it is changing (36, 6.61)

Produce the help package, text, data fields, images, etc. in the appropriate medium (48, 6.50)

Determine key words for searching online help (51, 6.44)

Add "hotspots" (buttons) to topics to provide linkages (62, 5.94)

Appendix B: Problems in the Design of Online Help

Note: We allowed participants to add problems for each step and to include that problem in the ranking for that step. However, very few participants did this and no given problem was added by more than one person. We did not include these additional problems in the appendix, but the range of rankings given a problem occasionally reflects the additional problems included in the rankings of one or more people. That is, a problem may show a range of rankings from one to ten when there are only nine problems for that step.

Research and Review

Product is still under development. No documentation exists, and functional spec is still evolving. (median=2, range=1-5)

Tracking down information. There's a lack of centralized information about the project. It's quite possible that no one has explicitly identified and documented information about resources, constraints, intended users, etc., even if the application software is virtually complete. (3, 1-5)

Dealing with incomplete, incorrect, or unspecific specification of requirements for the online help system (i.e., no written requirements, no user interface mockup, no end-user involvement). (3, 1-6)

Obtaining information about users. Trouble identifying and gaining access to the intended user. It may be that no one really knows who the application is for, so we hear about five or six different audiences. (4, 1-6)

Making sure that all team members have a full understanding of the system, users, documentation, etc. (5, 1-6)

Gaining access to people in software/hardware. (5, 1-7)

Project Management

Accurately estimating the resources required to develop online help. (median=2, Range=1-4)

Achieving coordination and closure. Getting team members to agree to schedule. Balancing responsibilities for this project with other priorities. (3, 1-5)

Adapting to change. Communicating problems and changes and adjusting schedules as needed. (3, 1-5)

Working with engineering. They don't deliver software on time or meet deadlines. Or, they work on the wrong part of the program: You get to the point where the program is 90% there and, of course, all that's lacking is the human interface—the part that your documentation has to describe. (3.5, 1-5)

Getting software developers and their management to appreciate the importance and difficulty of developing a good online help system. They tend to view doing the software for online documentation as the lowest priority project of all. They don't appreciate what's needed to develop an effective online help system—they think it's something you tack on to the product in a few day's work. (4, 1-5)

Create Design Specifications for the Content

Differentiating between online help and documentation. Convincing others (especially the software developers and managers) that the online help material should not be (a) simply a re-hash or shorter version of what's in the manual (and organized the same way) or (b) completely organized around the structure of the application (menus, commands, screen fields, etc.). (median=2, range=1-9)

Dealing with an requirements definition that is incomplete or incorrect because user/customer needs were not understood. (3, 1-8)

Being as familiar as possible with the software and still maintaining a novice user perspective. (4, 1-8)

Adjusting to changing functionality requirements for the application. Ensuring that the proper level (current update) of info is available at the time the help files are created. (4, 1-7)

Getting all of the people involved in a project to agree on the same content and sign off in a timely manner. (5, 1-9)

Counteracting the tendency of the team to regard online help as a reference, i.e., look up a term, get a definition. (6, 1-9)

Using the audience profile as a standard for carefully deciding what to include and what to omit. Since few audience profiles are narrow and uniform, you probably will need to satisfy a number of different audiences. Moreover, you will need to ensure that the help system satisfies a user as he or she progresses from novice user to expert. (6, 3-8)

Getting to have input into what goes into the content and how it is organized. Programmers usually dictate the system content and it is usually organized according to the way the system functions. (7, 2-9)

Educating novice writers. (8, 4-10)

Create Design Specs for the Interface and for Functionality.

Understanding how the intended audience for the online help will actually use it. Knowing how and when it will be used. (median=2, range=1-4)

Choosing a model to work from and keeping it simple. Selecting an appropriate "look & feel" for the online help interface. (3, 1-8)

Matching your ideal interface (what you'd like to do based on research and your user analysis) with what you have to do (limitations imposed by product/platform). (4, 1-8)

Keeping design specs consistent with the interface for the software so that users don't get lost in the online help and become even more discouraged. Getting the help application to be functionally consistent with the application. (4, 1-7)

Getting to have input into what interface should look like. The interface is often cast in concrete by a team of engineers who can only imagine online help like the help they saw on mainframes—very rigid, very mechanical, and very much a reference rather than a series of procedures. So you get locked into a very limited idea of help, because of the paucity of links and the lack of imagination in the interface you are provided. Interface is usually designed around system functionality and is not intuitive for the end user. (4 1-8)

Overcoming subjective opinions about what is good. (5, 1-8)

Finding a designer who's capable of coming up with an interface simple enough to learn quickly, flexible enough to handle advanced functionality, and consistent with the basic metaphor(s) of the application. (6, 1-8)

Educating brand new writers. (7, 3-8)

Prototyping

Getting the time allocated to permit formal prototype evaluation. (median=3, range=1-8)

Getting the time allocated to permit prototype development. (3, 1-8)

Finding people with sufficient expertise to plan and conduct effective evaluation and testing of the prototype. (4, 1-10)

Finding an appropriate (easy-to-use) tool to build and evaluate the prototype. (5, 1-10)

Getting funding allocated for development of the prototype. (5, 1-9)

Getting funding allocated for formal evaluation of the prototype. (5, 1-10)

Gaining access to the users. (6, 1-10)

Dealing with suggested changes. As soon as people look at the prototype, they want a new interface. (6.5, 1-10)

Getting technical support—i.e., getting some techie type to devote some time to coming up with the prototype (not easy with everyone's schedules around here). (7, 1-10)

Getting development to "sign up" for the prototype and reserve screen/function key etc. "real estate" for you in your help system. (8, 1-10)

Produce Help Text

Writing clearly and concisely, writing understandable text that enables the user to solve the problem, but doesn't provide so much information that it's overwhelming or too much to read. (median=2, range=1-7)

Adjusting to a changing application interface or functionality that forces changes in the information or user interface design of the system. (3, 1-5)

Determining the content. Inability to predict what information the user is really looking for. (3, 1-8)

Meeting Deadlines. (3.5, 1-8)

Keeping track of the latest version of the help text and making sure that it's complete. (5, 1-8)

Transforming all of the documentation into the form required. (5.5, 1-8)

Assuming that there is a separate technical documentation group, keeping in touch with this group and ensuring that you can have access to their files for importation into the help system. (5.5, 4-8)

Avoiding the temptation to write new material. (7, 4-8)

Implement Help

Deciding how to link the help. (median=3.5, range=1-11)

Dealing with technical problems involving window system capabilities and display handling. (4, 1-11)

Adjusting to strange operating system quirks that affect the way that the online help system works and prevent the design from being implemented as specified. (4, 1-10)

Making sure that help files are in the correct form so that they can be ported from word processing or development systems into the product form. (4, 1-11)

Obtaining sufficient programmer resources. (5, 2-10)

Coordinating between traditional documentation, SQA, and development. Ensuring that the help remains consistent with the application and the documentation. (6, 1-10)

Dealing with a lack of a sufficiently rich interface between the application and the help system (i.e., the application may not pass enough "current state" information to the help system). (6, 1-10)

Dealing with build problems caused by insufficient unit testing . (7, 1-11)

Getting agreement on specs. (7, 2-11)

Using the right number of graphics. Too many, and you have no room for all the help. Too few, and the screens look too dense. (7.5, 4-9)

Finding appropriate tools. (8, 4-11)

Pre-release Testing/Quality Assurance

Keeping abreast of prerelease changes in the application and documentation. The product code may not yet be frozen. (median=3, range=1-7)

Testing early enough that it's not too late to revise the design in a thoughtful way and to implement the changes suggested by testing. (4, 1-5)

Getting the needed technical, time, and human resources. (4, 1-10)

Knowing when online help is good enough. Demonstrating compliance with stated requirements or just knowing when to quit. (4, 1-9)

Coming up with a testing plan. (5, 1-11)

Getting QA resources dedicated to testing the help subsystem,
including content, and not just testing the functionality of the application proper. (7, 1-9)

Coping with lack of expertise in testing. (7, 3-11)

Avoiding or coping with tedium. It is tedious to ensure that each piece of help text and each help facility works from version to version, and that tedium can prevent complete testing. (8, 1-11)

Avoiding the tendency to use testing time as development "slip" time. (8, 2-11)

Communicating with testers (e.g., ensuring that the testers know to get their comments, suggestions, and problems back to the writers). (8, 2-11)

Getting everything (room, camera, subjects, etc.) together. Logistics. (9, 3-11)

Post-release Testing and Maintenance

Getting quality information (i.e., decent information that tells you enough that you can make judgements based on it) from the field and from service about what the problems are and what priority should be assigned to fixing them. The "Customer Feedback Void." (median=1, range=1-3)

Assigning responsibility for monitoring updating once the team has finished the project. Individuals will have moved on to other assignments and might even have left the company by the time the update is going to be developed. Management loses interest. Nobody ever seems to have the energy for monitoring and updating that they had for the original release. (2, 1-3)

Setting constraints on revisions. Updates are usually maintenance releases, where stuff has to get out the door pretty quickly. So you can't keep revising the help system ad infinitum. (3, 1-3)

Online Help Systems: An Approach to Summative Evaluation

by

Thomas M. Duffy
Indiana University

Ann Aaron
Hewlett Packard

James Palmer
Apple Computer

7 September 1989

This work was funded in part by the United States Army, Human Engineering Laboratory, Aberdeen Proving Grounds, MD under contract # DAAA1-86-K-0019-AC85.

Online Help:
An Approach to Summative Evaluation

Thomas M Duffy — Indiana University
Ann Aaron — Hewlett Packard
James Palmer — Apple Computer

The overall goal of the work under this contract has been to aid the design and development process by providing both a conceptual orientation and specific tools. This segment of the work deviates from the overall theme: our focus here is on the end user. That is, we will provide a specific tool and a rationale for conducting a consumer oriented (summative) evaluation of online help systems. Furthermore, we will provide evaluation data on more than 20 help systems for commercial products.

We don't expect the average consumers to actually conduct the evaluation. In general they would not have the expertise or the time required. However, we do hope that representatives of the consumer -- software reviewers -- will make use of the tool and thus make the results of the evaluation available to the user. The result, if that goal is realized, will be an index of the usability of help systems that consumers can use as part of their own assessment of a package. The evaluation data we provide on help systems is meant to serve as the foundation for this comparative database.

The evaluation system we developed is called the Help Design Evaluation Questionnaire. An HDEQ evaluation provides a reasonable level of detail -- there are 52 questions in eight categories -- and thus we expect that the results will also inform developers. Developers will be able to compare their product to the product of competitors, to earlier versions of their product or to other systems in our database. In the best of all possible worlds, this usability index could be part of the advertisement for particularly well designed products. But, it must be kept in mind that our primary goal is to provide a tool for the consumer to use in assessing usability of the help system.

In the following paragraphs we will provide an overall rationale for the design of HDEQ. Following that we will present the evaluation data along with evidence for reliability of the instrument. The HDEQ, itself, is presented in Appendix A.

Goals for the Design

In beginning this effort, we identified several criteria for the design which we felt were essential if it was to meet our overall objective of aiding the consumer. Those criteria were:

Efficient. It is simply a fact of life that few people are wiling or able to spend the time or the money on a lengthy, detailed evaluation unless it is the evaluation of a feature that is system-critical for them (or for their audience, in the case of reviewers). Thus, if we expect the questionnaire to be used, it must be inexpensive to apply and require minimal time.

User Oriented. A user will only pay attention to evaluation data if he sees the relevance to his own need or use of the system. This means that a single score won't do. A single score is simply too general and vague for the user to interpret in terms of her context. What is required is a set of evaluation scores that address user relevant issues -- the tasks of the user and the features that will make those tasks easy or difficult.

Comparative. A set of scores can only be interpreted relative to other help systems. A score, whether it is a subjective rating or an objective tallying of characteristics only takes on meaning relative to what is possible. We will never have the "perfect" help system; there will always be features that we can criticize. The way to give the less than perfect scores meaning is to develop a database on a wide range of help systems, especially systems that are widely used and hence likely to be familiar to the consumers.

Demands only general skill and tools. If the evaluation is to be used by a wide spectrum of consumers or their representatives, than application of the questionnaire cannot require special expertise. We can not expect the evaluator to have particular technical knowledge, design knowledge, or rhetorical knowledge. That is, we can't anticipate that the evaluator is a programmer, a designer, or a rhetorician. If the evaluation calls for special skills in those areas, then we must either provide procedural information or, in the case of judgements, examples that can serve as reference points. We should emphasize, however, that this does not mean that anyone can go out and evaluate a help system. There will most certainly be a requirement for broad experience and expertise as an end user.

Valid. The evaluation instrument must have both face validity and predictive validity. Face validity is essential for user acceptance -- it simply won't be used if it does not appear to be assessing relevant attributes. Because the primary goal is conducting the evaluation is to help consumers identify useable help systems, the system must predict those systems or system features that will will be more or less difficult to use.

Reliable. The evaluation tool should yield the same relative score and diagnosis of a help system regardless of which help designer uses it.

Evaluating Content vs Design

In developing the HDEQ we found it useful to distinguish between the design of the help system and the details of the content. The content issues include the accuracy and the completeness of the information presented. We find the evaluation of these aspects of the help system to be particularly troublesome. Accuracy and completeness can only be evaluated by a detailed review of the application program. That is, the only way to determine if the information is accurate and complete is to identify the tasks that are done with the application and have a user perform those tasks in the application.

In essence, the evaluation of accuracy and completeness of the information seems to present significant problems in achieving our criteria of efficiency and comparability. Conducting a detailed analysis of the application program and conducting user tests to determine if the information for a task is adequate is a very time consuming job. Thus the efficiency problem. Furthermore, by tying the evaluation so closely to the use of the application we are faced with a very difficult problem of separating the influence of the application on the design of the help.

We have chosen to focus our energy on the development of an instrument to evaluate the design of the help system. The design includes the functionality, the graphic design, and the text design. It also includes the decision as to the categories of information to be presented. This is not information specific to the application, but rather the general categories of information that would be considered for inclusion in any help system, e.g., syntax, examples, etc. These are all features that can be evaluated without a detailed analysis or use of the application program. Of course, the evaluator will have to have some familiarity with the application -- it is simply that he will not be required to have a detailed understanding. Furthermore, the design features are less influenced by the nature of the application program. Note, we are not suggesting that they are not influenced by the application; of course they are. But we think it is possible to take those influences into account in the evaluation process.

The work reported here focuses on the evaluation of the *design* of the help system. Work on the *content* evaluation remains for the future. However, before turning to the design let us offer some thoughts on the content evaluation strategy.

Content Evaluation -- Some Thoughts

Evaluating the content of the online help involves determining if the information is adequate to support the user's performance of tasks in the application. That is, we are assessing the accuracy and completeness of the information. Completeness, however, can be thought of in two ways: the breadth of coverage and the depth of coverage. In terms of breadth, we want to know if the wide range of tasks a user might undertake are addressed in the help system. The depth of information is whether or not the information on a task is adequate to guide the user through his impasse.

For our purposes depth of information and accuracy of the information are very similar. That is, whether the information is incomplete or wrong it will keep the user from successfully completing the task. Furthermore, accuracy and completeness can only be assessed through user performance.

Evaluating the Design of Online Help

Structure for HDQS

The first step in designing the evaluation was to determine a structure. That is, what sort of summary information do we want to report to consumers? What subscores to we want to present? The approach we developed focuses on the user's tasks. This is consistent with the evaluation goal, stated earlier, of relevance to the user. Thus the we set as a goal, evaluating how the help system aids the user in:

- formulating the problem
- accessing the help
- selecting a help topic
- navigating between help topics
- scanning the help information
- understanding the information
- addressing the particular problem in a way the user can apply it
- transferring the information to the impasse in the application.

Selecting items

Given that structure, we next had to determine the evaluation method. Basically, our own analysis and the findings of Root and Draper (1983) suggest that a questionnaire would be most appropriate. The Root and Draper (1983) findings also suggest that we want to ask checklist type of questions about features of the help system.

Our strategy, therefore, was to develop a series of checklist questions that would assess the usability of the help system in accomplishing each user task. We reviewed help systems, interface, and technical manual design guidelines to identify potential evaluation items. This procedure resulted in a very large list of guidelines. Inclusion of them all would make it infeasible to actually use the evaluation questionnaire, so we attempted to restrict inclusion in four ways.

• **Discriminability**. We examine 10 help systems for commercial software to identify those features that are distinguishable between systems. In essence, we wanted to identify design issues that make a difference.

• **Similarity.** We attempted to consolidate those guidelines that were closely related into a single guidelines or evaluation item.

• **Feasibility.** We only addressed design features that one could reasonably expect in commercial help systems. Thus for example, we did not include guidelines for natural language processing or other features of intelligent help systems.

• **Relevancy.** We only looked at guidelines that could be clearly interpreted in terms of our evaluation goal. That is, we are attempting to assess the usability of the help system for accomplishing each of the user tasks outlined above. We only included guidelines where violation of the guideline would clearly lead to the system being less usable for accomplishing one of those tasks.

Relevancy: A rational approach.

The issue of relevancy requires more explanation. Particularly, we want to be clear about the criteria (or rationale) we used in determining relevancy. The goal in the design of a help system is to minimize the time and effort required to get the information needed to resolve an impasse. Ideally, we would test a user on the help system and measure how long they spent on each task. Of course we can't do that because of the time involved and because of the confounding effect of the application. Nonetheless, the evaluation goal is to determine how well the design of the help system minimizes time and effort.

In general we used results from the research literature and a rational analysis in judging whether or not a particular guideline was relevant to the time and effort required of the user in these tasks. The rational analysis was based in part on common sense and in part on the GOMS model (Card, Moran, and Newell 1983) of the information processing system as it relates to time and effort. Card et al (1983) postulate that three processing systems are relevant to the individuals interaction with the environment: perceptual, cognitive, and motor. Most every action (gross behavior) an individual takes, requires a response or an activity in each processing system. That is, the user must register the stimulus, link the stimulus to the response, and execute the response. For example, selecting an item from a menu requires identifying each menu item (perceptual), evaluating the relevance of the item (cognitive), notifying the motor system to respond (cognitive), and moving the cursor and making the selection (motor).

Following this approach, we examined the guidelines to determine how they affect the number of perceptual, cognitive, and motor processing tasks the user must engage in to successfully complete the larger task of accessing, navigating, etc. The GOMS approach does not address one time factor that is very relevant to our evaluation: system response time. Clearly, if a system is to provide information efficiently, it must respond promptly. There cannot be long waits while files are opened or assembled. Thus, we also system response time when conducting the rational analysis.

In sum, we want to include design features in the questionnaire that we can identify a priori as affecting the processing or the system response time. Inclusion of a design feature in the questionnaire will be rationalized based on the hypothesized increase or decrease in the amount of processing or wait time. How might a design feature increase the amount of processing? There are two possible ways: the minimum number of processing activities may be high or the demands on working memory may be high .

Design will vary in terms of the minimum number of processing tasks the user must perform to get and apply the information even when the user knows precisely where to go for the information and how to get there. The differences may be due to the number of stimuli that must be processed, e.g., the number of menu items that must be searched will be affected by the size and organization of the menu; the number of motor tasks and the system response time will be affected by the size of the menu hierarchy. It may also occur due to the complexity of the decision involved in determining the appropriateness of an action, e.g, the number of inferences a user must go through in deciding if a particular alternative is the one that will provide the help needed. Finally, the number of tasks may very as a function of the number of keystrokes required to make a response, e.g., the availability of word completion.

The working memory demands will increase under at least three conditions. First, systems that require recall rather than recognition place a greater load on working memory. This situation will arise anytime the expected alternative is not visible. This could occur in accessing help (having help visible vs having to guess where it might be) or in making a selection (command entry vs menu selection) .

The second working memory issue is overloading the capacity by requiring the user to remember too much. The working memory demands may increase when the user has complex navigation requirements and must remember information across several screen while also making navigation decisions or when a lot of information must be transferred from one screen to another (e.g., help information isn't visible while it is being applied ion the application).

Finally, the processing demands on working memory will be affected by the familiarity of the information. This may happen in the presentation of complex information. It may also occur when the menu items don't match expectations. In either case, the user must search long term memory pulling things into working memory and make inferences in order to find a match or an interpretation.

In summary, the "relevancy" criterion for selecting guidelines to be included in the questionnaire was based on a hierarchical information processing task analysis. At the top level are the major tasks the user must perform in using help. At the second level are the design features that affect the number of information processing activities the user will have to engage in in order to complete the major task.

The Help Design Evaluation Questionnaire.

Sorting the guidelines based on the four criteria and rephrasing them for use in a questionnaire resulted in the generation of 52 items for inclusion in the questionnaire. These items all ask the rater to either note the presence of a feature (e.g, the visibility of access to help when the person is in the application) or to rate the quality of a feature (e.g., the comprehensibility of the help information). The complete HDEQ is presented in Appendix A.

Administering HDEQ

HDEQ is an analytic tool. Administration of HDEQ calls for the analysis of the help system being evaluated. Thus, for example, in analyzing the menus, the evaluator is required to examine all menus in the help system.

This approach can be contrasted to an experience based evaluation, where the evaluator rates the help system based on her experience in using it. Under this approach, it would be important that evaluators have a common experience base before they rate a help system. The experience base of the evaluator will strongly influence the rating. However, because of the analytic nature of HDEQ, there is no need for the evaluator to have a specified prior experience. Administering HDEQ does require a general understanding of the application program and knowledge of the command set. However, the very process of evaluation will provide the necessary experience with the help system.

The HDQS should be administered by an individual with broad end user experience. The evaluator will have to navigate through all parts of the help system in conducting the evaluation and, without broad experience, the navigating will likely be frustrating and also lead to incomplete coverage. Additionally, a number of the questions do call for judgements. While we try to provide guidance for making those judgements, a breadth of experience will be very helpful.

HDEQ Evaluation Data

We used the Help Design Evaluation Questionnaire (HDEQ) to evaluate the online help for 25 commercial applications. We had three goals in conducting these evaluations. First, and foremost, we wanted to gather data on the reliability of our instrument. Thus, we had two individuals independently review 16 help systems and we assessed the degree to which their evaluations agreed. If they did not agree, then the instrument would not be very useful. Secondly, we wanted to conduct a formative evaluation of the HDEQ. In essence, we wanted to identify the weak spots in the system. We examined the correlations between raters on particular subsections of the HDEQ and we collected feedback from the reviewers. We hope to use this data, in the future, to improve the wording of the questions and the guidance we provide. Finally, if the HDEQ was reliable, we saw the evaluations as contributing the the development of a database. That is, individuals could use HDEQ to compare new help systems with the ones in our database.

While we evaluated a total of 28 help systems, only 16 of the systems were evaluated by both reviewers. The systems were distributed across main frames (4), MS DOS (13), and Macintosh (11) applications. They included operating systems (4), word processors/desk top publishing (7), graphics and presentational (3), data analysis/management (9) and system utilities (5). Interestingly, we evaluated 3 versions of help for Microsoft Word: 3.0 for MS DOS, and 3.0 and 4.0 for the Macintosh. Thus we will be able to compare design across systems as well as evaluate improvements in the new version of the system.

The overall ratings by each rater are presented in Table 1. We present the systems evaluated by both reviewers first, ordered by the overall rating. Those systems only evaluated by one or the other individual are presented at the bottom of the Table. Given the competition between Apple and IBM, we presume a large number of readers will attempt to calculate the difference in ratings between IBM and Apple applications. Such a comparison is not particularly meaningful since we made no attempt to obtain comparable IBM and Apple applications. However, we are sure that the comparisons will be made nonetheless and therefore let us note that for those systems where we had two raters, the mean rating for the Macintosh help systems is .54 while that for the IBM help systems is .45. The other cross platform comparison possible is between Microsoft Word for the IBM and the Macintosh, where we find the Macintosh 3.0 version rated higher than the IBM 3.0 version (.53 and .44 respectively) while the 4.0 IBM version is the highest rated of all. We might suggest from this that progress in technology and in attention to the help systems is much more important than the particular platform a help system is delivered on.

Application	Rater 1	Rater 2	Average Rating
Macintosh			
Pagemaker 3.0	.72	.65	.69
HyperCard 1.2	.61	.75	.68
PowerPoint 1.0	.56	.63	.60
Microsoft Word 3.0	.53	.63	.58
Filemaker Plus	.51	.53	.52
Adobe Illustrator 1.1	.57	.44	.51
Excel 1.0	.39	.48	.44
Quark Express 1.1	.33	.47	.40
Image Studio 1.0	.40	.38	.39
IBM PC			
Microsoft Word 4.0	.72	.67	.70
Microsoft Word 3.0	.44	.46	.45
Lotus 1-2-3 2.01	.42	.48	.45
PlanPerfect 3.0	.33	.51	.42
DataPerfect 2.0	.45	.36	.41
WordStar 4.0	.40	.37	.39
WordPerfect 4.1	.36	.33	.35
Macintosh			
Hyperscan 1.0	.78		
Applelink 4.0	.67		
IBM PC			
Excel 2.0		.67	
DBase III		.63	
Systat 3.0		.46	
FoxBase 1.21		.42	
Epsilon 2.01		.29	
HP DeskManager B.03	.47		
Operating Systems			
Andrew		.78	
VMS		.43	
Tops-20		.41	
Unix		.33	

Table 1. Overall rating of the design of help system using HDEQ. Scores may range from .00 to 1.00, with larger scores indicating a more effective system.

Table 2 presents the interrater reliability (Pearson product moment correlation) and the range of scores for the 16 help systems for which there were two raters. Reliabilities and ranges are provided for both the overall score and for the eight subscores representing the eight user tasks. A reliability coefficient of r=.73 was obtained for the overall score. That is, the raters registered a significant and reasonable amount of agreement in their ratings. We would hope that with improvements in the system we could raise the reliability to around .80 or perhaps somewhat higher. Given the diversity of the systems to be evaluated and the necessary subjectivity of some of the judgements we would never expect to reach the reliability of .90+ found with objective tests on a narrow topic. In essence, the current reliability suggests that we can make gross but not fine discriminations between systems.

User's Task	Interrater Reliability	Range of Mean Rating
Formulate Problem	.69	.00-1.00 (1.00)
Access Help	.63	.28-.84 (.56)
Select Topic	.42	.29-.82 (.53)
Scan Information	.48	.38-.89 (.51)
Comprehend Information	.24	.45-.94 (.49)
Obtain type of information needed	.68	.20-.70 (.50)
Navigate	.81	.10-.87 .67
Transfer to the Application	.62	.06-.82 (.76)
Total Score	**.74**	**.35-.70 (.35)**

Table 2. The Pearson product moment correlation between raters and the range of mean scores across help systems, for both the overall and the subtask HDEQ ratings.

The range of scores on each of the subtasks was reasonably high. Indeed, the scores covered the entire possible range for the task of formulating a problem. The range for all of the other subtasks was between .49 and .76. Thus, the assessment led to a spread of scores across at least half of the scale. This all suggests that the assessment instrument can discriminate between help systems as to the degree to which they aid the user in each of the tasks.

Turning now to the reliability scores, we see that there is a wide range of reliability coefficients. In particular, the data suggest that reliability is unacceptably low for the measurement of how well the system aids the user in selecting a topic, in scanning information, and especially in comprehending the information. We are not particular surprised that the scanning and comprehension subscores had the lowest reliability. The questions in each of these categories are almost all subjective and hence a personal judgement. However, we are disappointed that the reliabilities are as low as they are. Future work in the development of HDEQ will involve the generation of examples that can serve as guideposts for each alternative for each question in these areas. The examples will help to establish a common framework between raters.

Tables 3 through 10 present the scores for each system on each of the 8 parts of the HDEQ. Again, the 16 systems evaluated by both raters are presented first. Table 2 presents the summary information on the data in these Tables. The specific data serves simply to contribute to the data base that may be used in comparing help systems in the future.

Application	Rater 1	Rater 2	Average Rating
Macintosh			
Pagemaker 3.0	1.00	1.00	1.00
HyperCard 1.0	1.00	1.00	1.00
PowerPoint 1.01	.25	.25	.25
Microsoft Word 3.0	.00	.75	.38
Filemaker Plus	.75	.50	.63
Adobe Illustrator 1.1	.50	.00	.25
Excel	.00	.00	.00
Quark Express	.00	.00	.00
Image Studio	.00	.00	.00
IBM PC			
Microsoft Word 4.0	1.00	.75	.87
Microsoft Word 3.0	.25	.25	.25
Lotus 1-2-3	.25	.00	.13
PlanPerfect	.00	.50	.25
DataPerfect	.25	.00	.13
WordStar	.00	.00	.00
WordPerfect 5.0	.00	.00	.00
Macintosh			
Hyperscan 1.0	.75		
Applelink 4.0	.25		
IBM PC			
Excel		.67	
DBase III		1.00	
Systat		.00	
FoxBase		.00	
Epsilon		.25	
HP Desk Manager v. B.03	.75		
Operating Systems			
Andrew		1.00	
VMS		.25	
Tops-20		.00	
Unix		.25	

Table 3. Rating of the degree to which the design of help system attempts to capture the way the user may formulate the problem. Scores may range from .00 to 1.00, with larger scores indicating a more effective system.

Application	Rater 1	Rater 2	Average Rating
Macintosh			
Pagemaker 3.0	.22	.33	.28
HyperCard 1.0	.56	.67	.62
PowerPoint 1.01	.44	.67	.55
Microsoft Word 3.0	.45	.67	.56
Filemaker Plus	.44	.67	.56
Adobe Illustrator 1.1	.56	.67	.62
Excel	.56	.67	.62
Quark Express	.44	.67	.56
Image Studio	.56	.67	.62
IBM PC			
Microsoft Word 4.0	.78	.89	.84
Microsoft Word 3.0	.22	.33	.28
Lotus 1-2-3	.56	.67	.62
PlanPerfect	.45	.89	.67
DataPerfect	.56	.66	.61
WordStar	.56	.33	.45
WordPerfect 5.0	.45	.44	.45
Macintosh			
Hyperscan 1.0	1.00		
Applelink 4.0	.78		
IBM PC			
Excel		1.00	
DBase III		.44	
Systat		.78	
FoxBase		.66	
Epsilon		.66	
Hp Desk Manager B.03	.56		
Operating Systems			
Andrew		.44	
VMS		.33	
Tops-20		.44	
Unix		.33	

Table 4. Rating of the degree to which the design of help system aids the user in accessing help. Scores may range from .00 to 1.00, with larger scores indicating a more effective system.

Application	Rater 1	Rater 2	Average Rating
Macintosh			
Pagemaker 3.0	.69	.54	.62
HyperCard 1.0	.62	.69	.66
PowerPoint 1.01	.83	.80	.82
Microsoft Word 3.0	.60	.50	.55
Filemaker Plus	.57	.60	.59
Adobe Illustrator 1.1	.60	.38	.49
Excel	.60	.70	.75
Quark Express	.70	.80	.75
Image Studio	.77	.46	.62
IBM PC			
Microsoft Word 4.0	.69	.67	.68
Microsoft Word 3.0	.40	.36	.38
Lotus 1-2-3	.62	.69	.66
PlanPerfect	.62	.60	.61
DataPerfect	.57	.00	.29
WordStar	.62	.77	.70
WordPerfect 5.0	.39	.50	.45
Macintosh			
Hyperscan 1.0	.80		
Applelink 4.0	.90		
IBM PC			
Excel		.46	
DBase III		.69	
Systat		.50	
FoxBase		.38	
Epsilon		.00	
HP Desk Manager B.03	.39		
Operating Systems			
Andrew		.77	
VMS		.38	
Tops-20		.30	
Unix		.00	

Table 5. Rating of the degree to which the design of the help system aids the user in selecting a topic. Scores may range from .00 to 1.00, with larger scores indicating a more effective system.

Application	Rater 1	Rater 2	Average Rating
Macintosh			
Pagemaker 3.0	.85	.82	.84
HyperCard 1.0	.62	.82	.72
PowerPoint 1.01	.58	.92	.75
Microsoft Word 3.0	.69	.69	.69
Filemaker Plus	.46	.62	.54
Adobe Illustrator 1.1	.92	.69	.81
Excel	.30	.54	.42
Quark Express	.31	.46	.38
Image Studio	.42	.50	.46
IBM PC			
Microsoft Word 4.0	.85	.92	.89
Microsoft Word 3.0	.58	.77	.68
Lotus 1-2-3	.62	.46	.54
PlanPerfect	.39	.69	.54
DataPerfect	.69	.47	.58
WordStar	.54	.46	.50
WordPerfect 5.0	.86	.67	.77
Macintosh			
Hyperscan 1.0	.92		
Applelink 4.0	.85		
IBM PC			
Excel		.54	
DBase III		.92	
Systat		.85	
FoxBase		.69	
Epsilon		.31	
HP Desk Manager B.03	.64		
Operating Systems			
Andrew		.69	
VMS		.77	
Tops-20		.69	
Unix		.77	

Table 6. Rating of the degree to which the design of the help system aids the user in scanning the help text. Scores may range from .00 to 1.00, with larger scores indicating a more effective system.

Application	Rater 1	Rater 2	Average Rating
Macintosh			
Pagemaker 3.0	.77	.56	.67
HyperCard 1.0	.55	.60	.58
PowerPoint 1.01	.58	.35	.47
Microsoft Word 3.0	.50	.42	.46
Filemaker Plus	.46	.50	.48
Adobe Illustrator 1.1	.67	.70	.69
Excel	.27	.27	.27
Quark Express	.25	.41	.33
Image Studio	.21	.35	.28
IBM PC			
Microsoft Word 4.0	.77	.62	.70
Microsoft Word 3.0	.33	.60	.47
Lotus 1-2-3	.33	.42	.38
PlanPerfect	.25	.36	.31
DataPerfect	.17	.35	.26
WordStar	.38	.21	.30
WordPerfect 5.0	.21	.18	.20
Macintosh			
Hyperscan 1.0	.80		
Applelink 4.0	.77		
IBM PC			
Excel		.75	
DBase III		.50	
Systat		.45	
FoxBase		.50	
Epsilon		.17	
HP Desk Manager B.03	.34		
Operating Systems			
Andrew		.87	
VMS		.42	
Tops-20		.50	
Unix		.42	

Table 7. Rating of the degree to which the design of the help system aids the user by presenting the right type of information. Scores may range from .00 to 1.00, with larger scores indicating a more effective system.

Application	Rater 1	Rater 2	Average Rating
Macintosh			
Pagemaker 3.0	.83	.73	.78
HyperCard 1.0	.72	.94	.83
PowerPoint 1.01	.67	.88	.78
Microsoft Word 3.0	.72	.94	.83
Filemaker Plus	.56	.72	.64
Adobe Illustrator 1.1	.94	.94	.94
Excel	.50	.61	.56
Quark Express	.12	.78	.45
Image Studio	.89	.89	.89
IBM PC			
Microsoft Word 4.0	.83	.78	.81
Microsoft Word 3.0	.56	.78	.67
Lotus 1-2-3	.56	.61	.59
PlanPerfect	.61	.50	.56
DataPerfect	.57	.93	.75
WordStar	.67	.66	.67
WordPerfect 5.0	.67	.33	.50
Macintosh			
Hyperscan 1.0	1.00		
Applelink 4.0	.83		
IBM PC			
Excel		1.00	
DBase III		.67	
Systat		.29	
FoxBase		.71	
Epsilon		.36	
HP Desk Manager B.03	.39		
Operating Systems			
Andrew		.86	
VMS		.79	
Tops-20		.71	
Unix		.21	

Table 8. Rating of the degree to which the style in which the help text is written facilitates understanding the help information. Scores may range from .00 to 1.00, with larger scores indicating a more effective system.

Application	Rater 1	Rater 2	Average Rating
Macintosh			
Pagemaker 3.0	.50	.62	.56
HyperCard 1.0	.80	.93	.87
PowerPoint 1.01	.36	.31	.34
Microsoft Word 3.0	.62	.46	.54
Filemaker Plus	.31	.38	.35
Adobe Illustrator 1.1	.23	.15	.19
Excel	.39	.57	.48
Quark Express	.44	.30	.37
Image Studio	.23	.15	.19
IBM PC			
Microsoft Word 4.0	.50	.46	.48
Microsoft Word 3.0	.36	.47	.42
Lotus 1-2-3	.39	.62	.51
PlanPerfect	.23	.31	.27
DataPerfect	.20	.00	.10
WordStar	.15	.31	.23
WordPerfect 5.0	.23	.31	.27
Macintosh			
Hyperscan 1.0	.60		
Applelink 4.0	.87		
IBM PC			
Excel		.62	
DBase III		.54	
Systat		.30	
FoxBase		.15	
Epsilon		.43	
HP Desk Manager B.03	.46		
Operating Systems			
Andrew		.64	
VMS		.23	
Tops-20		.40	
Unix		.40	

Table 9. Rating of the degree to which the design of the help system aids the user in navigating through the help system. Scores may range from .00 to 1.00, with larger scores indicating a more effective system.

Application	Rater 1	Rater 2	Average Rating
Macintosh			
Pagemaker 3.0	.86	.63	.75
HyperCard 1.0	.00	.38	.19
PowerPoint 1.01	.75	.88	.82
Microsoft Word 3.0	.63	.62	.63
Filemaker Plus	.50	.25	.38
Adobe Illustrator 1.1	.12	.00	.06
Excel	.50	.50	.50
Quark Express	.37	.37	.37
Image Studio	.12	.00	.06
IBM PC			
Microsoft Word 4.0	.86	.25	.56
Microsoft Word 3.0	.25	.13	.19
Lotus 1-2-3	.00	.37	.19
PlanPerfect	.13	.25	.19
DataPerfect	.63	.50	.57
WordStar	.25	.25	.25
WordPerfect 5.0	.13	.25	.19
Macintosh			
Hyperscan 1.0	.38		
Applelink 4.0	.12		
IBM PC			
Excel		.62	
DBase III		.25	
Systat		.50	
FoxBase		.25	
Epsilon		.13	
HP Desk Manager B.03	.25		
Operating Systems			
Andrew		1.00	
VMS		.25	
Tops-20		.25	
Unix		.25	

Table 10. Rating of the degree to which the design of the help system aids the user in transferring the help information back to the application. Scores may range from .00 to 1.00, with larger scores indicating a more effective system.

Summary and Conclusion

The goal of this work was to develop an instrument that can be used in the summative evaluation of online help systems. The goal was to design a system that is focused on the user, provides a means for comparing help systems, is valid and reliable, is efficient to use, does not require special skills or tools. The resulting evaluation method is a rating system. It focuses on the design of the help system rather than the accuracy and completeness of the information. The particular emphasis in looking at the design, is the degree to which it aids the individual in obtaining and applying information.

The Help Design Evaluation Questionnaire (HDEQ) consists of 52 items that are distributed across eight tasks. The tasks are the eight tasks involved in using help. The help system receives an overall score as well as a score for the degree to which it supports each of the eight user tasks. Scores range from 00 to 1.00.

The HDEQ was applied to 28 help systems with sixteen of those systems being evaluated by two independent raters. Total evaluation scores ranged from .35 to .70 indicating that the HDEQ did lead to a reasonable discrimination between systems. Additionally, the interrater reliability of .74 suggests that the discriminations are reliable. An examination of the subscores for the eight tasks indicated that all subareas resulted in a reasonable spread in scores. However, the reliability was unacceptably low for the scores indicating the ease of comprehension, scanning the text, and selection of a topic. Future work will look to develop examples of each alternative for each question in these areas so that the examples can provide a more common frame for evaluators.

In conclusion, this work has resulted in the development of a reliable and valid tool for evaluating online help systems. The database formed by the 28 help systems we evaluated can provide the benchmark for assessing the effectiveness of new help systems.

In addition to refining the HDEQ, future work will be aimed at developing a companion instrument for evaluating the accuracy and completeness of the help information.

Appendix A

Help Design Evaluation Questionnaire (HDEQ)

A Design Evaluation Instrument

developed by

Thomas M. Duffy
Learning Resources
Indiana University
Bloomington, IN 47401

March 1, 1989

This work was funded in part by the United States Army Human Engineering Laboratory, Aberdeen Proving Grounds, MD under contract DAAA1-86-K-0019.

Overview

This instrument may be used to evaluate the design of any online help system. "Online help" in this context is a computer program that provides job-aiding information on the use of the application software. This evaluation instrument assumes that the online help information is accurate and complete.

Eight design features of a help system are evaluated with this instrument:

- support for problem representation
- ease of access to help
- ease of topic selection
- ease of scanning the help text
- appropriateness of the type of help information presented
- comprehensibility of the content
- ease of navigation within the help system
- ease of tranferring information from help to the application

The rating of each of these features is based on the amount of cognitive, perceptual, and motor effort that each segment of the system imposes on the user, e.g., how much work and time it takes to access the help system.

The questions focus on the weaknesses of the help system. The higher the raw score, the poorer the help system. We convert the raw score for each design feature into a "Component Score" that is positively related to quality, i.e., a high Component Score indicates that the particular component of the help is well-designed. The "System Score" is the average of the Component Scores, providing an overall rating of the help system. Each component is weighted equally in calculating the System Score.

Instructions

1. Complete the top of the Evaluation Summary on page 3.

2. Answer the questions in an Evaluation Category. Under each question, select the option that best describes the help system you are evaluating. Enter that number under "Your Score". **If the question does not apply to the help system that you are evaluating, skip it.**

3. When you have finished the evaluation in a Category, add the scores that you entered under "Your Score". Place that sum in the space for "Evaluation Score" at the top of the first page for that category.

4. Add the "Worst Scores" for the items you answered. **<u>Do not</u> include the worst scores for the items you skipped.** Place the sum in the space for "Worst Score" at the top of the first page for that category.

5. Calculate the Best Score that the software could have received in each category. The Best Score equals the number of questions that you answered in that category. Enter the Best Score at the top of the first page of that Evaluation Category.

6. When you have completed all of the evaluation items, transfer the Evaluation Score, Worst Score, and Best Score from the top of each evaluation category to Columns A (Evaluation Score), B (Worst Score Possible), and C (Best Score Possible) in the Evaluation Summary on Page 3.

7. Calculate the Component Score for each category and the System Score.

A well-designed component and a well-designed help system will have scores near 1.0. Use the Component Score and the System Score to compare help systems.

Evaluation Summary

Name:______________________________
Manufacturer:__________________________
System tested on:________________________

Version number:_______________________
Today's date:__________________________
Evaluator:____________________________

Enter the appropriate number in each cell.	A Evaluation Score	B Worst Score Possible	C Best Score Possible	D A minus C	E B minus C	F D divided by E	G Component Score 1.0 minus F
Prob Rep							
Access							
Menus							
Format							
Content							
Compreh.							
Navigation							
Linkage							

To determine the System Score:

1. Sum column G.

2. Divide by 8 to get the System Score.

System Score:_______

Category I: Problem Representation

Does the help system support different naming for tasks and commands?

Worst Score:____ Best Score:____ Evaluation Score:____

	Worst Possible Score	Your Score
A. Are alternative menu systems available to reflect differing purposes when seeking help (e.g., alphabetical menus to support searching for a particular command; task-based menus that correspond to real-world tasks; task-based menus to reflect computing tasks; or expert and novice menu organizations)?		
1. Yes, alternative representations are available, and alternative organizations make a lot of sense. 2. Yes, alternative representations are available, but one or more of the alternative organizations does not appear to be very helpful. 3. There is only one menu system, but there are both tasks and commands on it. 4. There is only a command menu.	4	______
B. Does the help system enable the novice user to use familiar terminology in linking to the relevant help topics?		
1. An index or a keyword search system (e.g., a "find" function) is available that includes an extensive list of synonyms for the functions in the application program as well as a list of real-world tasks that the functions tend to be a part of (e.g., "cut" and "paste" commands can be accessed through the task listing "moving a paragraph"). 2. Either the tasks or synonyms (but not both) are part of a keyword search system or index; or, both tasks and synonym-based access is available but is very limited. 3. There is a glossary that defines terms in the application program in a manner that the novice user can understand. 4. The help system does not contain a glossary/index or a keyword search system.	4	______

Worst Score:____ Best Score:____ Evaluation Score:____	Worst Possible Score	Your Score
A. Where must you be to access help? 1. Anywhere in the application program. 2. Only in certain places, e.g., main menu. 3. Must first exit the application or current activity.	3	
B. How do you access help? 1. A continuously displayed and meaningful prompt. 2. A menu item that is not continuously displayed on the screen; or a continuously displayed prompt that is hard to interpret. 3. A command that is intuitively obvious or a "standard" but is not continuously visible (i.e., <F1> on an MS DOS program). 4. A command that is not intuitively obvious or traditional and is not continuously visible.	4	
C. How long does it take for help to appear on the screen after you enter the command for it? 1. Two seconds or less. 2. Three to eight seconds. 3. More than eight seconds.	3	
D. Does the help place you in a relevant subset of help? 1. Yes, placed in a relevant help based on current context (current mode) in the application program, user expertise, or user history. 2. No, placed in the same subset of help regardless of current context.	2	
E. Does the program prompt you to seek help when you commit errors within the application? 1. Yes. 2. No.	2	

Category III: Menus

How easy is it to locate information in the menus?

Worst Score:____ Best Score:____ Evaluation Score:____

	Worst Possible Score	Your Score
A. How many items are on the main menu? Consider the main menu to be the first menu that has a list of actual help topics from which the user may choose, <u>not</u> an introductory menu that offers the user navigational or "branching" choices (for instance, a menu which allows the user to choose help on commands, or help on tasks, would be a navigational menu, not a main menu). **(Skip this question if there are 15 or fewer items in the entire help system).** 1. 15 to 50. 2. 51 to 70; or 8 to 14. 3. More than 70; or fewer than 8. 4. There are no menus; just a keyword search system.	4	____
B. If there is a menu hierarchy (more than a main menu), what is the average number of items that are on the submenus (including embedded menus, e.g., words in the help text that are selectable)? **(Skip this question if there is no menu hierarchy).** 1. Between 7 and 15. 2. Fewer than 7 or more than 15.	2	____
C. Are all the items on menus visible at once? 1. Yes 2. No. In some menus, the user must scroll or page to see all menu items--but never for more than two screens. 3. No. In most menus, the user must scroll or page to see all menu items--but never for more than two screens. 4. No. In most menus, there are more than two screens of menu items.	4	____
D. How are higher level menus and submenus linked? **(Skip this question if there are no submenus).** 1. Submenus can be seen without leaving the higher level menu (e.g., through the use of pop-up windows or splitting the screen). 2. Submenus are included in the information or text on a help topic (e.g., through hypertext or as a list at the end of the help information). 3. Submenus replace higher level menus (that is, the menu selections are a path to help text rather than help being provided on all menu entries).	3	____

Category III: Menus
Continued

How easy is it to locate information in the menus?	Worst Possible Score	Your Score
E. How are the items on the menu organized? 1. Organization of the menu items is based on shared relevance (e.g., used in same task, part of the same topic,etc.) and there are headings for the groups. 2. Menus are grouped as in 1, but there are no headings for the groupings. 3. Items on the menu are presented alphabetically. 4. Menu items have neither an alphabetical nor a common use organization, e.g., they may be organized by frequency of use or the organizational structure may not be apparent.	4	____
F. How do you select menu items? 1. Click on the item with a mouse or, if there is not a mouse capability, type a letter or number associated with the item. 2. Move the cursor to the selection with full (up-down, left-right) cursor control. 3. Type the first letter or letters of the item to be selected. 4. Make the selection with the cursor without full cursor control (and the items are in more than one column).	4	____

Category IV: Format

Does the format of the help text facilitate searching for and understanding the needed information?

Worst Score:____ Best Score:____ Evaluation Score:____

	Worst Possible Score	Your Score
A. If the help text frequently takes two or three screens, but not more, how does the user move through the text? **(Skip this question if the help text never requires more than one screen or is almost always more than three screens).** 1. Page through. 2. Scroll.	2	____
B. If the individual help texts tend to be more than three screens in length, as in an online document, how do you move between screens? **(Skip this question if the help texts are almost always three screens or less in length).** 1. Page through; and there is a visible menu available to jump to particular subtopics. 2. Page through; but there is no menu available to jump to particular subtopics. 3. Paging is not available; the user must scroll through the text. There is, however, a menu available to support jumping to particular subtopics. 4. No paging or subtopic menu is available; the user must scroll through the text.	4	____
C. How are lists formatted? **(Skip this question if no lists are presented).** 1. Usually presented as a list with bullets, numbers, dashes, etc. 2. Usually presented as a list but with no bullets, dashes, numbers, etc. 3. Sometimes presented as a list 4. Almost never presented as a list but instead are presented as sentences or phrases in a paragraph.	4	____
D. How much of the screen appears to be given to margins and spacing between chunks of information? (Do not include spacing between lines in your estimate.) 1. <u>Looks like</u> at least 50%. 2. <u>Looks like</u> between 30% and 50%. 3. <u>Looks like</u> less than 30%.	3	____

Category IV: Format

Continued

Does the format of the help text facilitate searching for and understanding the needed information?	Worst Possible Score	Your Score

E. Are the different types of information (command-syntax, function, examples,etc.) clearly separated by spacing? **(Skip this question in the unlikely event that there is only one type of information in the help text).** — Worst Possible Score: 3 — Your Score: ____

1. Frequently.
2. Sometimes.
3. Infrequently (they are typically all mixed together.)

F. Are headings or some form of highlighting (such as underlining, boldface, or extra line spacing and indentation) used to identify the different types of information in the help text? **(Skip this question if there is so little information presented that highlighting is not necessary).** — Worst Possible Score: 4 — Your Score: ____

1. Almost always.
2. Frequently.
3. Occasionally.
4. Very seldom.

G. Are basic organization and format principles applied consistently across the different help texts? — Worst Possible Score: 4 — Your Score: ____

1. Yes, the appearance (layout, highlighting and organization) is highly structured and consistent from topic to topic.
2. Yes, there is consistency in the application of formatting principles, but few principles are applied (more structuring of the presentation would be helpful).
3. Formatting principles are applied but are not consistent from topic to topic.
4. No formatting principles are applied consistently.

H. Is the text easy to read (i.e., is the typeface large enough and legible)? — Worst Possible Score: 3 — Your Score: ____

1. Yes, very consistently.
2. Most text is easy to read but some information is presented in small or difficult-to-read fonts; or poor highlighting strategies make it difficult to read some text.
3. Most of the text is very difficult to read

I. Are both upper- and lowercase letters used? — Worst Possible Score: 2 — Your Score: ____

1. Yes.
2. No.

Category V: Content

What information is presented in the help text?

Worst Score:____ Best Score:____ Evaluation Score:____

	Worst Possible Score	Your Score
A. Is the content task-oriented? (Consider a task to be the job or action a user wants to complete when he or she consults the help system. Tasks may be described in the help text in terms of computer tasks, such as copying a file, or in terms of real-world tasks, such as creating a footnote in a word processor application). 1. Focus is on tasks and how to use the commands in completing those tasks. 2. Focus is on describing command syntax and function, but some attention is given to tasks or examples of applications. 3. Help only describes command syntax, function, and related information (e.g., bugs). An example of how to enter the syntax is given (there is no task/application information, or it is trivial.) Or, syntax is not relevant and the help system only describes the function of commands. 4. Help only describes syntax or function without giving an example of the way in which the syntax would be entered.	4	______
B. Is the help system interactive, asking you for information and using that to determine which help to present? 1. Yes, and it is well-done. 2. Yes, but it is not well done or consistently done. 3. No.	3	______
C. Are there levels of explanation (e.g. quick reference vs. elaborated; novice vs. expert; simplified vs. technical)? 1. Yes, and it is very complete and distinguishable 2. Yes, but it is not very complete or different 3. No.	3	______

Category V: Content

Continued

What information is presented in the help text?

D-J. For each of the following questions, enter the appropriate score if the feature is appropriate to the help system. **(Skip the question if the feature is not appropriate for aiding the use of the application program).**

	Absent but should be present.	Present for some help but should be in all.	Present as necessary in all help.	Worst Possible Score	Your Score
D. Is syntax information given?	2	1.5	1	2	______
E. Is function information given?	2	1.5	1	2	______
F. Is a list of related commands given?	2	1.5	1	2	______
G. Are possible applications suggested?	2	1.5	1	2	______
H. Is a concrete example of how the command is used presented in enough detail that you could imitate it?	2	1.5	1	2	______
I. Are bugs, warnings, and trouble-shooting advice given?	2	1.5	1	2	______
J. Are tutorials available from the help?	2	1.5	1	2	______

Category VI: Comprehensibility

How clearly is the help text written?

Worst Score:____ Best Score:____ Evaluation Score:____

	Worst Possible Score	Your Score
A. How easy is it to understand (not apply) the help information while browsing through it? 1. Almost always very easy to understand. 2. Often very easy to understand. 3. Often very difficult to understand. 4. Almost always difficult to understand.	4	____
B. Are sentences overly complicated in structure? 1. Almost always a simple structure. 2. Usually a simply structure. 3. Usually a complicated structure. 4. Almost always a complicated sentence structure.	4	____
C. Are sentences in passive voice? 1. Infrequently. 2. Sometimes. 3. Frequently. 4 Almost always.	3	____
D. Do lists have parallel structure? 1. Frequently. 2. Sometimes. 3. Infrequently. 4. Almost never.	3	____
E. Are noun strings used? 1. Infrequently. 2. Sometimes. 3. Frequently. 4. Almost always	3	____

Category VI: Comprehensibility

Continued

How clearly is the help text written?	Worst Possible Score	Your Score
F. Does the text refer to the user with personal pronouns (e.g., "you") or use the imperative (e.g., "Press the return key")? 1. Frequently. 2. Sometimes. 3. Infrequently.	4	
G. Is the vocabulary, beyond the use of command and task names, unnecessarily difficult or technical? 1. Almost always easy to understand. 2. Usually easy to understand. 3. Usually difficult to understand. 4. Almost always difficult to understand.	3	
H. Are functional graphics used? Functional graphics are used for informational purposes, not just for motivational or aesthetic reasons. **(Skip this question if no graphics are appropriate).** 1. Yes, graphics are used consistently. 2. Yes, but only occasionally. 3. No.	3	
I. Are functional graphics easy to understand? **(Skip this question if no graphics are appropriate).** 1. Very easy to understand. 2. Reasonably easy to understand in general. 3. Very difficult to understand.		

Category VII: Navigation

How easy is it to navigate between help texts?

Worst Score:____ Best Score:____ Evaluation Score:____

	Worst Possible Score	Your Score
A. How helpful is the information on how to use the help system? **(Skip this question if the system is so simple that information (help) on the help is not needed).**		
1. Very complete and clear— gives overview and substantive information on use and is readily available (that is, there is a prompt for accessing help on help and that prompt is visible whenever the user enters help).	5	____
2. Help on help is provided but does not describe the organization of the help or does not give strategies for use.		
3. The help on help is clear and complete but the user must leave the help system to read the help on help (e.g., when it is in a separate file).		
4. Help on help is not available within the help system and it does not describe the organization of the help system or suggest strategies for use.		
5. No help on help is provided.		
B. How helpful is the overview on the application software?		
1. Provides a conceptual model to assist in thinking about the software and gives hints on strategies and potential pitfalls.	4	____
2. Provides some useful detailed information but does not provide a general view that helps in thinking in a global way about the software.		
3. Does not provide useful detail or does not help in thinking globally or generally about the software.		
4. There is no overview on the application software.		

Category VII: Navigation

Continued

How easy is it to navigate between help texts?	Worst Possible Score	Your Score
C. How do you move around in the help system? **(Skip this question if the system has 15 or fewer commands).** 1. There is a network of submenus of help information that go directly to related topics or go to a menu of related topics. 2. There is a single menu but you can go from any one topic to another through either a keyword or a "see also" listing in the help without returning to the menu. 3. There is a single menu and you must use this to access all other help topics. 4. There are multiple menus, but there are only certain fixed paths through the menus or help database (e.g. you must return through a path of two or more items or menus before taking another route).	4	____
D. How long does it take to move from one help topic to another (within the same general topic area if the help is broken into topic areas)? 1. Less than 2 seconds. 2. 2 to 5 seconds. 3. More than 5 seconds.	3	____

Category VII: Navigation

Continued

How easy is it to navigate between help texts?

	Worst Possible Score	Your Score
E. What support is available for navigating? **(Skip this question if the system is so simple that it is impossible to get lost -- less than 15 entries or a single main menu that the user always goes through).**		
1. Four or more of the following navigation supports are available:	4	____
• bookmarking (allows the user to "mark a place" in the help system, and to return directly to that place from the application program at a later time)		
• navigation requirements that are always visible or the method for accessing the requirements is prompted		
• emergency procedures for recovering from getting lost, e.g, a " return to main menu" option)		
• a map of the help system showing your current location		
• the ability to preview a topic or see the context surrounding the discussion of a topic		
2. Two or three of the above five supports are available.		
3. One of the above five supports is available.		
4. None of the above five supports is available.		

Category VIII: Linkage

Does the link between the help system and the application software facilitate applying the help information to the users's tasks or problems?

Worst Score:____ Best Score:____ Evaluation Score:____

	Worst Possible Score	Your Score
A. Can you transfer help information to the application program (e.g., through cut and paste)? 1. Yes. 2. No.	2	____
B. Can you view the relevant portion of the application program while in the help system? 1. Yes, windowing permits viewing both the help and the application simultaneously. 2. Sometimes (there is an overlapping window with fixed placement). 3. Yes, but I must toggle between the help and application screen (that is, they cannot be seen at the same time, you must "switch" between them). 4. No, I must leave help to see the application.	4	____
C. Can you work on the application while help is on the screen? 1. Yes, for all of the help I need to see. 2. Yes, but only for a small portion of the help information I need to see. 3. No.	3	____
D. Does the functioning of the help system mimic the functioning of the application software (e.g. using the same movement commands, same kind of menu system)? 1. Yes, fully or almost fully mimics the application software. 2. Somewhat. 3. No.	3	____

References

Apple (1987) Human interface guidelines: The apple desktop interface. Menlo Park, CA: Addison-Wesley.

Apple Computer (1989) HyperCard stack design guidelines. Menlo Park, CA: Addison-Wesley.

Bethke, F., Dean, P., Kaiser, E., Ort, E. & Pessin, F. (1981). Improving the usability of programming publications. IBM Systems Journal, 20 (3), 306-320.

Bewley, W., Roberts, T., Schroit, D., Verplank, W. Human Factors testing in the design of the Xerox's 8010 "Star" office workstation. undated.

Brockmann, R. J. (1986). Writing Better Computer User Documentation: From Paper to Online. NY: John Wiley & Sons.

Smith, D. C., Irby, C., Kimball, R., Verplank, B., and Harslem, E. Designing the Star user interface. Byte, 1982, 242-243.

Card, S. K., Moran, T. P., & Newell, A. (1983). The psychology of human-computer interaction Hillsdale, NJ: Lawrence Erlbaum Associates.

Duffy, T.M. (1985). Industrial manuals: A matter of priorities. In D.H. Jonassen (Ed.), The Technology of Text. Vol. 2. Engelwood Cliffs, NJ: Educational Technology Publications.

Duffy, T., Gomoll, T., Gomoll, K., Palmer, J., & Aaron, A. (1988). Writing online information: Expert strategies (interviews with professional writers). Pittsburgh, PA: Communications Design Center.

Duffy, T.M., Ackerman, J., Grantham, D., and Kelly, '.. (1985) An analysis of the effectiveness of the documentation for the Texas Instruments controllers, minicomputer, and Professional. unpublished technical report, Pittsburgh PA: Communication Design Center, Carnegie Mellon University.

Elm, W. & Woods, D. (1985, October). Getting lost: a case study in interface design. Proceedings of Human Factors Society. Westinghouse Scientific Paper: 85-1C60-CONRM-P2.

Engle, S. & Granda, R. (1975, December). Guidelines for man/display interfaces. Technical Report 00.2720, IBM Poughkeepsie Laboratory.

Ericcson, K. A. and Simon, H. A. (1984) Protocol Analysis: Verbal reports as data. Cambridge, MA: MIT Press.

Gardiner M. M. and Christie, B. (Eds.), (1987) Applying cognitive psychology to user-interface design. New York, NY: John Wiley and Sons.

Good, M., Whiteside, J., Wixon, D., and Jones, S. (1983) Building a u: r derived interface. Proceedings of the CHI 83 conference on human factors in computing systems. New York: American Computing Machinery. (pp 1032-1043.

Haas, C. & Hayes, J. R. (1987). Effects of Text Display Variables on Reading Tasks: Computer Screen vs. Hard Copy. Technical Report CDC-3 (ERIC # 260 387), Pittsburgh, PA, Communications Design Center, Carnegie Mellon.

Hannafin, M. J. and Peck, K. L. (1988) The design, development, and evaluation of instructional software. New York: Macmillan Publishing Co.

Heines, J. M. (1984) Screen design strategies for computer-assisted instruction. Bedford, MA: Digital Press.

Gould, J and Lewis, C (1983) Designing for usablility -- key principles and what designers think. *Proceedings of CHI '83: Human Factors in Computing Systems*. Boston, MA: American Computing Machinery.

Jones, M. K. (1989) Human computer interaction: A design guide. Englewood Cliffs, NJ: Educational Technology Publications.

Kearsley, G. (1988). Online help: Design and implementation. Menlo Park, CA: Addison-Wesley Publishing Co.

Kruk, R. & Muter, P. (1984). Reading of continuous text on video screens. Human Factors, 26 (3), 339-345.

Muter, P., Latremouille, S.A., Treurniet, W.C., & Beam, P. (1982). Extended reading of continuous text on television screens. Human Factors, 24, 401-414

Norman, D. (1988) The psychology of everyday things. New York: Basic Publishing Co.

Norman, D. and Draper, S. (Eds.) (1986) User centered system design. Hillsdale, NJ: Lawrence Erlbaum and Associates.

Olson, D.(1985) On the designing and understanding of written texts. In T. M. Duffy and R. Waller (Eds.) Designing Usable Texts. New York: Academic Press.

Ong, W. (1982). Orality and literacy: The technologizing of the word. London: Methuen.

Redish, V. (1987) Writing for people who are "Reading to learn to do". Paper presented at the National Reading Conference, St. Petersburgh, FL.

Robertson, C.K. and Akscyn, R. (1982). Experimental evaluation of tools for teaching the ZOG frame editor. Technical Report. Pittsburgh, PA: Carnegie Mellon University.

Root, R. and Draper, S. (1983) Questionnaires as a software evaluation tool. Proceedings of CHI '83. Boston, MA: American Computing Machinery.

Schriver, K. 1989 Evaluating text quality: The continuum from text focused to reader focused methods. IEEE Transactions on Professional Communication, 32, 33-45.

Shneiderman, B. (1986). Designing the User Interface: Strategies for Effective Human-Computer Interaction. Reading, MA: Addison-Wesley.

Smith, S. and Mosier, J. (1986). Guidelines for designing user interface software. (ESD-Technical Report TR-86-278). Hanscom Air Force Base, MA: Electronic Systems Division, United States Air Force

Walker, J. (1987). Issues and strategies for online documentation. IEEE Transactions on Professional Communication, PC 30, 235-248.